

T.C.
EGE ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

**DERİN ÖĞRENME MODELLERİ KULLANARAK
KREDİ KARTI DOLANDIRICILIK TESPİTİ İÇİN
KARAR DESTEK SİSTEMİ GERÇEKLEŞTİRİMİ**

Peyman BAGHDADI

Danışman: Prof. Dr. Mustafa Serdar KORUKOĞLU

Bilgisayar Mühendisliği Anabilim Dalı Adı
Bilgisayar Mühendisliği Doktora Programı

İzmir
2024

Peyman BAGHDADI tarafından Doktora tezi olarak sunulan “**Derin Öğrenme Modelleri Kullanarak Kredi Kartı Dolandırıcılık Tespiti için Karar Destek Sistemi Gerçekleştirimi**” başlıklı bu çalışma EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliği ile EÜ Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi’nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve **8 Şubat 2024** tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

Jüri Üyeleri:

İmza

Jüri Başkanı : Prof. Dr. M. Serdar KORUKOĞLU

Raportör Üye : Dr. Öğr. Üyesi Ensar Arif SAĞBAŞ

Üye : Prof. Dr. Aybars Uğur

Üye : Prof. Dr. Hasan BULUT

Üye : Prof. Dr. Aytuğ ONAN



EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

ETİK KURALLARA UYGUNLUK BEYANI

EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Doktora Tezi olarak sunduğum “**Derin Öğrenme Modelleri Kullanarak Kredi Kartı Dolandırıcılık Tespiti için Karar Destek Sistemi Gerçekleştirimi**” başlıklı bu tezin kendi çalışmam olduğunu, sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi, tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı, bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

.... / / 20..

İmzası

Peyman BAGHDADI



ÖZET**DERİN ÖĞRENME MODELLERİ KULLANARAK KREDİ KARTI
DOLANDIRICILIK TESPİTİ İÇİN KARAR DESTEK SİSTEMİ
GERÇEKLEŞTİRİMİ**

BAGHDADI, Peyman

Doktora Tezi, Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Prof. Dr. Mustafa Serdar KORUKOĞLU

Şubat 2024, 76 sayfa

Teknolojideki hızlı gelişmeler günümüzdeki yaşamının her yönünü etkilemektedir. Bankacılık ve finansal ödemeler de bu gelişmelerden faydalanan ancak hizmet ettikleri kolaylıkların yanı sıra bazı dezavantajları da bulunmaktadır. En yaygın dolandırıcılık türü olarak bilinen kredi kartı dolandırıcılığı, bankacılık şirketlerinin, finans sektörlerinin ve bunlara bağlı müşterilerinin her yıl milyarlarca dolar kayba uğramasına neden oluyor.

Kredi kartı dolandırıcılık sorunlarını önlemek ve ortadan kaldırmak için birçok güncel makine öğrenme ve son zamanlarda derin öğrenme yöntemlerine dayalı çözümler sunulmasına rağmen, bunların çoğusu hız ve performans arasında bir denge kuramamaktadır. Temel sorunlar: konsept kayması, dengesiz veriler, büyük ölçekli veriler ve gerçek zamanlı tespit ile uğraşmakta olup bunların hepsini bir arada ele alınması neredeyse imkansız hale getirmektedir. Bunun yanı sıra bankaların ve finans sektörünün dolandırıcılık veri setlerini açıklama konusundaki, itibarlarının tehlikeye atılmasından dolayı isteksizlikleri, bu alandaki zorlukları daha da artmasına sebep olmaktadır.

Bu çalışmada, daha verimli ve uygulanabilir bir model oluşturmak için yukarıdaki zorlukları göz önünde bulundurarak topluluk sınıflandırıcı derin öğrenme modellerini kullanarak kredi kartı dolandırıcılığının tespitine yönelik bir karar destek sistemi önermekteyiz. Bir önyükleme maksimum oylama topluluğu modeli geliştirmek için temel sınıflandırıcılarımız olarak Kısıtlı Boltzmann makinesi ve Uzun Kısa Süreli Bellek kullanmaktayız. Nihai karar verme topluluk sınıflandırıcı modelimiz, RBM ve LSTM sınıflandırıcıları için %50 oy hakkını eşit olarak dikkate alınarak, ardından bunun sahte veya gerçek bir işlem olduğunu tahmin etmek için sonuçları normalleştirip birleştirmektedir.

Değerlendirme ölçütleri olarak AUC–ROC, AUC–PR, *Kesinlik*, *Geri çağırma*, F1–ölçüt, *karışıklık matrisi* ve geçen süreyi dikkate almaktayız. Gerçek dünyadaki Avrupa kart sahipleri veri seti üzerindeki deneysel sonuçlarımız, önerilen topluluk modelinin hız ve performans arasındaki denge açısından sahadaki güncel modeller arasında daha verimli olduğunu ortaya koyduğumuzu kanıtlamaktayız.

Anahtar sözcükler: Kredi kartı dolandırıcılığı tespiti, Derin öğrenme, Otomatik kodlayıcı, Uzun Kısa Süreli Bellek, Kısıtlı Boltzmann Makinesi, Topluluk sınıflandırıcı.



ABSTRACT**IMPLEMENTATION OF A DECISION SUPPORT SYSTEM FOR
CREDIT CARD FRAUD DETECTION USING ENSEMBLE DEEP
LEARNING MODELS**

BAGHDADI, Peyman

Ph.D. in Computer Engineering

Supervisor: Prof. Dr. Mustafa Serdar KORUKOĞLU

February 2024, 76 pages

Fast growth in technology affects all aspects of today's life. Banking and financial payments also profit from this progress, but besides the facilities they serve, there are some disadvantages. Credit card fraud, known as the most prevalent fraud type, causes billions of dollars in loss for banking companies, financial industries, and their related customers every year.

Although many solutions offer for preventing and eliminating credit card fraud based on up-to-date machine learning and, lately, deep learning methods to solve the problem, most of them do not strike a balance between speed and performance. The main problems are dealing with concept drift, imbalanced data, large-scale data, and real-time detection, which are very challenging, and handling them all together is almost impossible. Besides this, the disinclination of banks and financial industries to reveal their fraud dataset, which causes putting their reputation at stake, added more challenges in this field.

In this study, we propose a decision support system for credit card fraud detection using Ensemble DL models by considering the above challenges to build a more confident and applicable model. We utilize a Restricted Boltzmann Machine and Long Short-Term Memory as our base classifiers for developing a bootstrap max-voting ensemble model. Our final decision-making ensemble model considers %50 voting rights for RBM and LSTM classifiers equally, then normalize and aggregate the results to predict whether it is a fraudulent or genuine transaction. We consider AUC-ROC, AUC-PR, precision, recall, F1-measure, confusion matrix, and elapsed time as evaluation metrics. Our experimental results on the real-world European cardholder dataset reveal that the proposed ensemble

model is more efficient in terms of the balance between speed and performance, among the recent models in the field.

Keywords: Credit card fraud detection, Deep learning, Auto encoder, Long Short Term Memory, Restricted Boltzmann Machine, Ensemble learning.



ÖNSÖZ

Bu tez çalışmasında kredi kartı dolandırıcılık tespiti için, mevcut olan gerçek veri setin üzerinde araştırma yapıp, sahte işlemlerin tespit edilmesi amaçlanmıştır. Bu işlem Avrupa'da bir bankanın kart sahiplerinin 2013 yılında kayıt yapılan işlemlerin üzerinde gerçekleştirilmiştir. Banka sektöründe çalışma tecrübemden dolayı, kredi kartı dolandırıcılık ile ilgili yapay zeka bazlı modellerin geliştirilmesine ilgilenmiştim. Bu konuda çalışmaya da danışman hocam Prof. Dr. Serdar Korukoğlu tavsiyeleri üzerine karar verdim. İnsanların alışveriş davranışlarının üzerinden kredi kartı dolandırıcılığı tespit edebildiği çalışmaları incelemekten sonra, bu insanları nasıl dolandırılıp dolandırılmadıklarını merak ettim. Bu merak üzerine giderek tez çalışmamı şekillendirdim. Tez çalışmasının tam olarak anlaşılabilmesi için kredi kartı dolandırıcılık türlerine, tespit yaklaşımların adımlarına, makine öğrenme ve derin öğrenmesi konularında bilgi sahibi olunması gerekmektedir. Literatürde pek fazla sayıda gerçek zamanlı kredi kartı dolandırıcılık tespiti çalışması bulunmuyor olsa da çalışmamız ile kazandırmış olduğumuz gerçek zamanlı dolandırıcılığı tespit edilmesini amaçlayan, önceden bilinen ve bilinmeyen dolandırıcılık davranışı algılayan yaklaşım sayesinde bu alanda çalışacak olan araştırmacılara çok destek vereceğini düşünmekteyim.

İZMİR

8 Şubat 2024

Peyman BAGHDADI



İÇİNDEKİLER

ÖZET	vii
ABSTRACT	ix
ÖNSÖZ	xi
İÇİNDEKİLER	xiii
ŞEKİLLER DİZİNİ	xvi
ÇİZELGELER DİZİNİ	xix
SİMGELER VE KISALTMALAR DİZİNİ	xxi
1. GİRİŞ	1
2. GEÇMİŞ ÇALIŞMALAR	8
2.1 Makine öğrenme metodları ile KKDT	8
2.2 Derin öğrenme metodları ile KKDT	13
3. VERİ KÜMESİ	17
3.1 Veri Analizi	17
3.2 Veri setin önışlem yapılması	19
3.2.1 Veri setin öznitelik çıkarımı	20
3.2.2 Otomatik Kodlayıcı	211
3.2.3 Öznitelik Çıkarma İçin Otomatik Kodlayıcı Geliştirme	22
3.3 İkinci Veri Kümesi Oluşturma	24
4. YÖNTEM	26

4.1 Kısıtlı Boltzmann Makinesi	26
4.1.1 Kısıtlı Boltzmann Makinenin özellikleri	26
4.1.2 Kısıtlı Boltzmann Makinenin çalışma şekli	27
4.1.3 Gibbs sampling ve Contrastive Divergence	29
4.1.4 Kredi kartı dolandırıcılık için RBM Modeli geliştirme	30
4.2 Kısa Uzun Süre Bellek	33
4.2.1 LSTM çalışma şekli	33
4.2.2 Kredi kartı dolandırıcılık için LSTM model geliştirilmesi	36
4.3 Topluluk Sınıflandırıcılar	39
4.3.1 Bagging	40
4.3.2 Boosting	42
4.3.3 Stacking	43
4.3.4 Topluluk Sınıflandırıcı geliştirilmesi	45
5. DENEYSEL BULGULAR	48
5.1 RBM Modelinde Elde Edilen Deneysel Bulgular	48
5.2. LSTM Modelinde Elde Edilen Deneysel Bulgular	51
5.3 Topluluk Sınıflandırıcı ile Elde Edilen Deneysel Bulgular	54
6. GELİŞTİRİLEN KARAR DESTEK SİSTEMİNİN ÇALIŞMA PRENSİBİ	58
7. TARTIŞMA	60

7.1 RBM sonuçlarını diğer arařtırmalar ile karşılařtırma.....	60
7.2 LSTM sonuçları diğer arařtırmalar ile karşılařtırma	62
7.3 Tez Kapsamında Elde Edilen Bulguların Karşılařtırılması	63
8. SONUÇ.....	68
KAYNAKLAR DİZİNİ.....	70
TEŐEKKÜR	75
ÖZGEÇMİŐ	76

ŞEKİLLER DİZİNİ

Şekil 1.1. Tez çalışmasının temel akış şeması.	5
Şekil 3.1. Kırk Sekiz saat içerisinde işlem dağılımı.....	18
Şekil 3.2. Yirmi Dört saat içerisinde dolandırıcılık işlemlerin sıklığı.	18
Şekil 3.3. Her özelliğe göre gerçek ve dolandırıcılık işlem dağılımı.	19
Şekil 3.4. PCA ve Otomatik kodlayıcı tarafından doğrusal olmayan projeksiyonları.....	20
Şekil 3.5. Otomatik kodlayıcının genel yapısı.	21
Şekil 3.6. Otomatik Kodlayıcının Python ile geliştirilmesi.	23
Şekil 3.7. Eğitilmiş Otomatik kodlayıcın train ve test sonuçları.....	24
Şekil 3.8. Otomatik Kodlayıcının model yükleme ve ikinci veriset oluşturma Python kodu.....	24
Şekil 3.9. Geliştirilen veri yöntem şeması.....	25
Şekil 4.1. RBM modelin genel yapısı.	27
Şekil 4.2. RBM eğitimindeki ilk aşaması.....	27
Şekil 4.3. RBM öğrenmenin yeniden yapılandırma aşaması.	28
Şekil 4.4. Öğrenme sürecin gerçek ve tahminde bulunan dağılım örneği.....	29
Şekil 4.5. Her iki veri kümesi için RBM sınıflandırıcısının şeması.....	32
Şekil 4.6. Python ile geliştirilmiş RBM modeli.	32
Şekil 4.7. LSTM yapısının detayı.....	34
Şekil 4.8. LSTM'ın unut kapısının detayı..	34

Şekil 4.9. LSTM'ın giriş kapısının detayı..	35
Şekil 4.10. LSTM'ın çıkış kapısının detayı.	36
Şekil 4.11. LSTM için keras kütüphanesi kullanımı..	36
Şekil 4.12. LSTM modelini python kodu ile geliştirilmesi.	37
Şekil 4.13. Her iki veri kümesi için LSTM sınıflandırıcısının şeması..	39
Şekil 4.14. Bagging sınıflandırıcı topluluğunun genel şeması (ProjectPro, 2023)	41
Şekil 4.15. Boosting sınıflandırıcı topluluğunun genel şeması (ProjectPro, 2023)	43
Şekil 4.16. Stacking sınıflandırıcı topluluğunun genel şeması.	44
Şekil 4.17. Topluluk sınıflandırıcı modelin genel şematiği.	46
Şekil 4.18. LSTM ve RBM modelin çağrı ve tahminde bulunmaları Python kodu	47
Şekil 4.19. Topluluk sınıflandırıcı modelin Python kodu.	47
Şekil 5.1. Normal ve dolandırıcılık işlemlerin RBM model ile dağılımı.	48
Şekil 5.2. RBM ile elde edilen birinci ve ikinci veri setlerin ROC eğrileri.	49
Şekil 5.3. RBM ile elde edilen birinci ve ikinci veri setlerin PR eğrileri.	49
Şekil 5.4. LSTM ile elde edilen birinci ve ikinci veri setlerin ROC eğrileri.	52
Şekil 5.5. LSTM ile elde edilen birinci ve ikinci veri setlerin PR eğrileri.	52
Şekil 5.6. Birinci veri setin, RBM, LSTM ve topluluk modellerin AUC-ROC eğrileri.	54

Şekil 5.7. İkinci veri setin, RBM, LSTM ve topluluk modellerin AUC-ROC eğrileri	54
Şekil 5.8. Birinci veri setin, RBM, LSTM ve topluluk modellerin AUC-PR eğrileri.	55
Şekil 5.9. İkinci veri setin, RBM, LSTM ve topluluk modellerin AUC-PR eğrileri.	55
Şekil 6.1 Geliştirilen karar destek sistemin çalışma prensibi.....	58



ÇİZELGELER DİZİNİ

Çizelge 3.1. 227845 numune üzerinde eğitilmiş sonuçlar.....	23
Çizelge 5.1. RBM’ın birinci ve ikinci veri setin kesinlik, geri çağırma ve f1-ölçüt sonuçları.....	50
Çizelge 5.2. RBM’ın birinci ve ikinci veri setin karışıklık matrisi sonuçları...	51
Çizelge 5.3. RBM’ın işlem süresi sonuçları.	51
Çizelge 5.4. LSTM’ın birinci ve ikinci veri setin kesinlik, geri çağırma ve f1-ölçüt sonuçları.....	53
Çizelge 5.5. LSTM’ın birinci ve ikinci veri setin karışıklık matrisin sonuçları..	53
Çizelge 5.6. LSTM’ın işlem süresi sonuçları.	53
Çizelge 5.7. Birinci veri setin modeller arası kesinlik, geri çağırma ve f1-ölçüt sonuçları.....	55
Çizelge 5.8. İkinci veri setin modeller arası kesinlik, geri çağırma ve f1-ölçüt sonuçları.....	56
Çizelge 5.9. Birinci veri setin karışıklık matrisin sonuçları.	56
Çizelge 5.10. İkinci veri setin karışıklık matrisin sonuçları.	56
Çizelge 5.11. Topluluğun işlem süresi sonuçları.....	57
Çizelge 7.1. RBM’ın elde edilen sonuçların diğer araştırmalar ile kıyaslaması.	60
Çizelge 7.2. LSTM’ın elde edilen sonuçların diğer araştırmalar ile kıyaslaması	62
Çizelge 7.3. Topluluk sınıflandırıcı sonuçların diğer araştırmalar ile kıyaslaması.	63

Çizelge 7.4. Topluluk sınıflandırıcı karışıklık matrisin sonuçlarını diğer araştırmalar ile kıyası. 66

Çizelge 7.5. Modellerin işlem süresi karşılaştırması.....67



SİMGELER VE KISALTMALAR DİZİNİ

ML	Machine Learning – Makine Öğrenme
DL	Deep Learning – Derin Öğrenme
KKDT	Kredi Kartı Dolandırıcılık Tespiti
RBM	Restricted Boltzmann Machine – Kısıtlı Boltzmann Makinesi
LSTM	Long Short Term Memory – Kısa Uzun Süreli Bellek
ECH	European Card Holder – Avrupa Kart Sahibi (veri set)
AE	Auto Encoder – Otomatik Kodlayıcı
CD	Concept Drift – Kavram Kayması
SD	Skewed Distribution – Çarpık Dağılımı
RO	Rastgele Orman
LR	Lojistik Regresyon
KA	Karar Ağacı
SVM	Support Vector Machine
SVM-RFE	SVM - Recursive Feature Elimination
CCRI	Credit Card Risk Identification
KNN	K Nearest Neighbor – K En Yakın Komşu
ANN	Artificial Neural Network – Yapay Sinir Ağları
NB	Naive Bayesian
SMOTE	Synthetic Minority Oversampling Technique
TWB	Transaction Window Bagging – İşlem Penceresi Torbalama
SOM	Self Organization Map
CART	Classification And Regression Tree
UCSD	Universitesi of California San Diego
AIRS	Artificial Immune Recognition System
MDSA	Maliyete Duyarlı Sinir Ağları
UCI	University of California Irvine
MCC	Matthews Correlation Coefficient
YAO	Yanlış Alarm Oranı
PMCM	Prudential Multiple Consensus Model
FDS	Fraud Detection System

İBSA	İleri Beslemeli Sinir Ağı
HODA	Homojenlik Odaklı Davranış Analizin
RFM	Recency-Frequency-Monetary
DİA	Derin İnanç Ağı
MSE	Mean Squared Error
CD	Contrastive Divergence - karşılaştırmalı sapma
MCMC	Markov Chain Monte Carlo
RNN	Recurrent Neural Network - Tekrarlayan Sinir Ağı
S	Standart sapma
$\bar{X}$	Aritmetik ortalama
T	Toplam
Q_2	Medyan
S^2	Varyans
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
AUC-ROC	Area Under the ROC Curve
AUC-PR	Area Under the Precision Recall Curve
RL	Representation Learning

1. GİRİŞ

Teknolojideki hızlı büyüme, günümüz yaşamının tüm yönlerini etkilemektedir. Bunun gelişmesi ile birlikte elektronik ticaret ve online para işlemi kolaylaşmıştır ve kredi kartı bu alanda önemli bir rol oynamaktadır. Özellikle son yıllarda tüm dünyaya yayılan covid 19 virüsünün ardından kullanımının önemi çarpıcı bir şekilde artmaktadır. Bankacılık ve finansal görevleri kolaylaştırmayı kazandığımız avantajların yanında bazı dezavantajlar da bulunmaktadır. Bunlardan birisi, bir kişinin kredi kartı veya banka kartının ya da benzeri bir ödeme aracının, sahtekarlıkla para veya mülk elde etmek için yetkisiz kullanımıdır ve “kredi kartı dolandırıcılığı” olarak adlandırılmaktadır (FBI, 2023). Dolandırıcılık taahhüdünün arkasındaki temel sebebi, yasadışı yollarla yanlış zeminde kazanç elde etmektir. Bunun ekonomi, hukuk ve hatta insani ahlaki değerler üzerinde dramatik bir etkisi olmaktadır. Dolandırıcılık alanı genel olarak dört bölümden oluşmaktadır (Cherif et al. 2023):

- **Banka dolandırıcılığı:** En temel dolandırıcılık türü olup ortalama %50 üzerindeki dolandırıcılıklar bu tipten oluşmaktadır. Daha önce tanımını açıkladığımız gibi, iki çeşit: Kredi Kartı dolandırıcısı ve Para Aklama şeklinden oluşmaktadır.
- **Sigorta dolandırıcılığı:** %30 civarında dolandırıcılık payını alarak, ikinci temel dolandırıcılık türüdür. Bunlar sağlık, otomobil, ev ve iş yeri, ürün sigortası olarak farklı çeşitlerden oluşmaktadırlar.
- **Telekomünikasyon dolandırıcılığı:** telekomünikasyon hizmetlerinden veya sistemlerinden mali kazanç veya diğer kötü amaçlarla yararlanan yasa dışı faaliyetleri ifade eder. Bu dolandırıcılıklar genellikle telekomünikasyon ağlarının, ekipmanlarının veya hizmetlerinin yetkisiz kullanımını içerir. SMS dolandırıcılık, en yaygın örneği olarak sayılmaktadır. Bu tarz dolandırıcılık ortalama %10 oran bir paya sahiplerdir.
- **İnternet pazarlama dolandırıcılığı:** Gündemde farklı sosyal medya araçların üzerinde satış ve pazarlama çok yaygın hale gelmiştir. Bu tarz dolandırıcılıklar ortalama %5 üzerinde bir orana sahip olup oranı artmaktadır.

Para ve hizmetleri içeren, tüm teknolojik sistemler, dolandırıcılık eylemleriyle tehlikeye girebilirler ve yukarıdaki açıklamalara göre, banka dolandırıcılığın ne kadar önemli olduğunun kanıtına varmaktayız. Kredi terimi,

para olmadan mal alıp satma yöntemini tanımlamak için kullanılır. Kredi kartı dolandırıcılığı, kredi kartı veya banka kartı gibi bir ödeme kartı kullanılarak gerçekleştirilen dolandırıcılık için kapsamlı bir terimdir. Amaç, mal veya hizmet elde etmek veya bir suçlu tarafından kontrol edilen başka bir hesaba ödeme yapmak olabilir. Kredi kartı dolandırıcılığı sorunu, bankacılık şirketleri, finans sektörleri ve ilgili müşterileri için her yıl milyarlarca dolar zarara neden olmaktadır (Cherif et al. 2023; Mehndiratta et al. 2019). Bunun önlenmesi ve ortadan kaldırılması için kredi kartı dolandırıcılık tespiti alanına ihtiyaç vardır.

Dolandırıcılığa karşı koruma mekanizması iki temel çözümden oluşmaktadır:

- **Önlemek:** Bu çözümde dolandırıcılık ve benzer davranışların bir sisteme girmesini engellemeye çalışmaktadır. Çeşitli alanlardaki dolandırıcılık faaliyetlerini önlemek için tasarlanmış bir dizi araç, teknoloji ve süreçten oluşmakta olup, temel olarak güvenlik duvarı ve şifreleme/şifre çözme yaklaşımlardan faydalanmaktadır.
- **Tespit etmek:** Bu çözüm, aslında sistemde bir dolandırıcılık aktivite olup olmadığını algılamaya ve tespit etmeye çabada bulunan bir yöntemdir.

Dolandırıcılık tespiti yaklaşımında, sisteme girdikten sonra dolandırıcılık faaliyetlerini keşfetmeye ve tanımlamaya çalışır ve bunları bir sistem yöneticisine bildirilmektedir. Manuel yöntemler olarak adlandırılan kredi kartı dolandırıcılık tespiti için kullanılan temel ve eski teknikler, kişisel deneyimlere ve istatistiksel yollara dayanmaktadır. Geleneksel yönetimlerde alanın uzman kişi, gözlemci olarak tecrübelerini kullanarak, dolandırıcılığı tespit etmekte olup, şüphelendiğinde müşterileri arayıp işlemlerini onaylamaktadır. Örneği gece zamanlarda işlem yapılması, tam dolu, günlük maksimum miktarında işlem yapılması, yaşayan coğrafya bölgesinde hariç bir bölgeden bir işlem gelmesi gibi benzer davranışlar, şüpheli olarak tanımlanmaktadır.

Kredi kartı dolandırıcılık tespiti alanı bir işlemin sahte veya gerçek olup olmadığının kanıtlamak için, üzerinde çalışmalar yapılan bir alandır. Kredi kartı dolandırıcılığı tespitinde karşılaştığımız temel zorluklar ve sorunlar (Cherif et al. 2022; Shahana et al. 2023):

- **Kavram kayması:** Temel modelin zaman içinde değiştiği olgusunu ifade eder.

- **Çarpık dağılımı:** Dengesiz sınıf olarak da adlandırılır ve sahte örneklerin sayısı, normal olan örneklerle göre çok az olan bir durumdur.
- **Büyük miktarda veri:** Büyük ölçekli ve yüksek boyutlu dolandırıcılık veri seti ve çok sayıda özelliğin varlığıdır.
- **Gerçek zamanlı algılamayı desteklenmesi:** Algılama sürecinin verimli bir şekilde çalışmasını sağlamak için sınırlı zaman ve bellek ile ilgili çalışmadır.

Son yıllarda, araştırmacılar, yapay zekanın gelişmesi ve ilerlemesi ve makine öğrenme, veri madenciliği ve son zamanlarda derin öğrenmenin gündemde kullanılmasıyla, kredi kartı dolandırıcılık tespiti için otomatik yöntemler dayalı çalışma yapmaktalar. Makine öğrenmeye dayalı kredi kartı dolandırıcılık tespitinin üç türü vardır (Mehndiratta et al. 2019):

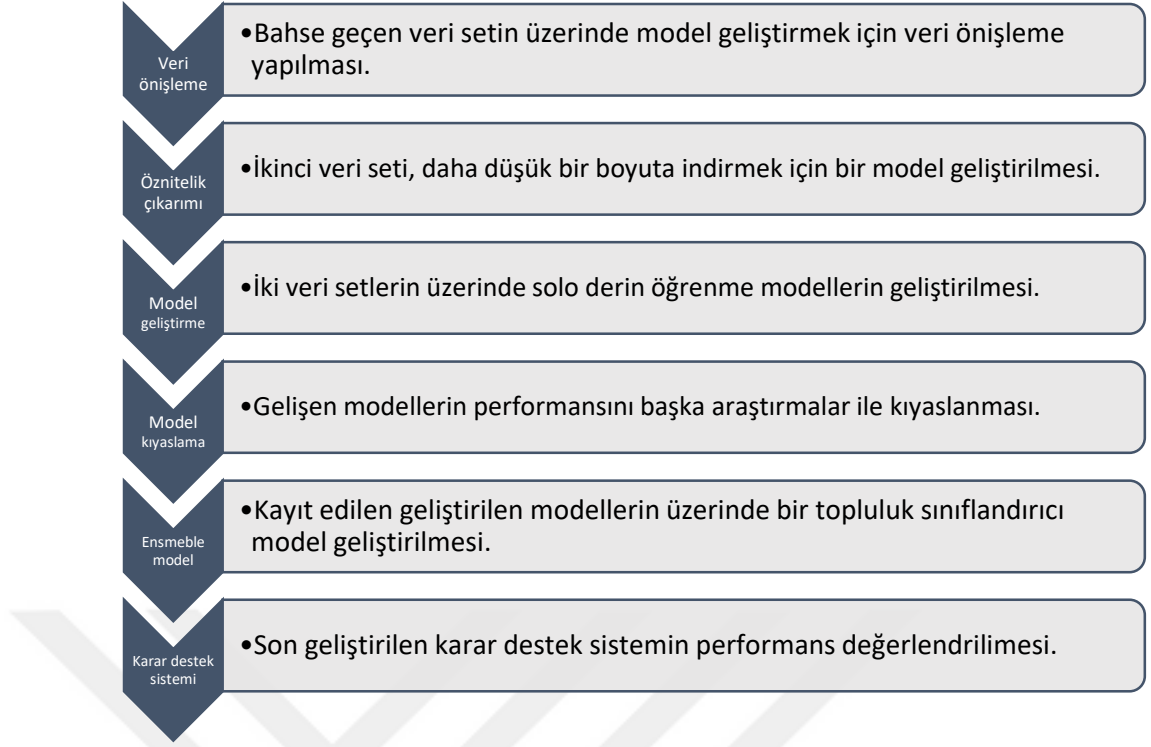
- **Gözetimli:** Gözetimli yöntemlerde, etiketli veri kümelerini kullanmakta olup, test veri kümesini sahte ve gerçek türlere ayıran bir model oluşturmaya çalışmaktadır.
- **Gözetimsiz:** Gözetimsiz yöntemlerde, test veri setlerini farklı gruplara ayırmaya ve her bir bireysel davranış kalıbını modellemeye çalışmaktadır ve son olarak benzer gruplarda yer almayanları veya aykırı değer olarak adlandırılanları işaretleyip sahte türler olarak kabul etmektedir.
- **Yarı gözetimli:** Yarı gözetimli ise her iki yöntem arasında yer almaktadır. Hem etiketli hem de etiketsiz verilerden bir sınıflandırıcı eğitmeye çalışıp normdan herhangi bir sapma olup olmadığını izlemekte olan bir modeldir.

Makine öğrenimi tabanlı kredi kartı dolandırıcılık tespiti kapsamında araştırarak, makine öğrenmesi mekanizmalarından hiçbirinin her açıdan en iyi olmadığını söyleyebiliriz. Bu nedenle sonuçlar, alan problemi ve değerlendirme performansı açısından incelenmesi ve yorumlanması gerekmektedir. Gözetimli tabanlı algoritmalar içinde, Rastgele Orman yapısına benzer modeller, daha iyi performansa sahip olan modellerdir (Afriyie et al. 2023), ancak Gözetimsiz algoritmalarla kıyaslayınca K-Ortalama daha iyi performans sergilemektedir. Ancak, bu araştırmaların sonuçları (Mittal et al. 2019), sınıf dengesizliği, kavram kayması, büyük boyutluk, gerçek-zaman desteği gibi kredi kartı dolandırıcılık tespit sistemlerinin sorunlarının, tek veya birkaç tanesi için çözüm sunmaktadır. Özetle, kredi kartı dolandırıcılık tespit sorunlarının tüm yönlerini içeren eksiksiz bir çözüm olmadığını söyleyebiliriz.

Daha sonra bilim adamları, makine öğrenme tekniklerini karıştırıp topluluk sınıflandırıcı yöntem olarak kredi kartı dolandırıcılık tespit problemlerini çözmek için kullanmakta olup, solo yöntemlerden daha iyi sonuç aldıklarını kanıtlamışlar (Carcillo et al. 2021). Algoritmik gelişmeler ve derin öğrenme tabanlı algoritmaların oluşturulmasıyla bu sonuçlar daha da iyi hale gelmektedir. Tüm değerlendirme kriterlerinde en iyi sonuçları veren derin öğrenme tabanlı kredi kartı dolandırıcılık tespiti için birçok araştırma yapılmıştır (Alarfaj et al. 2022).

Tüm bu yöntemler sorunu çözmeye çalışır, ancak makine öğrenmesinden ziyade derin öğrenme yöntemlerinin etkinliği ve üstünlüğü tamamen açıktır. Fakat Derin öğrenme yöntemleriyle yapılan tüm çalışmalarda, değerlendirme sonuçları, makine öğrenme modeller ile karşılaştırılmış veya aynı yapıya sahip derin öğrenme mimari ile kıyaslanmıştır. Diğer yandan, tüm odaklar model performansı üzerinde olup, çevrimiçi işlemler için fazla çaba ve ilgi gösterilmemiştir. Bu nedenle, derin öğrenme yöntemlerine dayalı olarak çevrimdışı ve çevrimiçi işlemler arasında bir denge bulmaya ihtiyaç duyulmaktadır.

Bu tez çalışmasındaki amaç, çevrimdışı ve çevrimiçi işlemler arasında bir denge bulmak için en iyi derin öğrenme tabanlı yöntemleri kullanmak ve bunları bir araya getirmektir. Tasarlanmış olan yeni model, kredi kartı dolandırıcılık tespitinin hızını ve performansını artmış ve son teknoloji derin öğrenme modelleriyle karşılaştırılmıştır. Daha önce bahsedilen kredi kartı dolandırıcılık tespiti ile ilgili temel zorluklara ve sorunlara dikkat edilerek, bunların her birisini çözmek için farklı çözümler sunulmuştur. Bu bakıştan bu alanda yeni çalışmalara öncülük edecek bir araştırmadır. Tezin temel akışı Şekil 1.1’de sunulmaktadır. Ayrıca ilgili tüm araştırmalarda inceleme yaparak, Avrupa Kart Sahibi veri setinin çok çeşitli araştırmalarda kullanıldığını ve bir şekilde yaygın bir eğitim seti metriği olduğunu öğrenmekte olup, bu nedenle bu veri setini kendi araştırmamızda da kullanılmasına karar verilmiştir.



Şekil 1.1. Tez çalışmasının temel akış şeması.

Tez çalışmasının temel katkıları şu şekilde özetlenebilir:

- **Mevcut veri setin üzerinde önleme yapılması:** Bankaların ve finans kuruluşlarının işlemlere ilişkin dolandırıcılık veri setlerini ortaya çıkarma konusundaki isteksizliklerine dikkat edilerek, kullanılabilecek sınırlı sayıda hazırlanmış veri seti bulunmaktadır. En yaygın olarak kullanılan veri setlerinden biri ECH veri seti'dir. Bu veri kümesine <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud> adresinden erişilebilmektedir. Bu veri setin üzerinde hem model geliştirmek için hem öznitelik çıkarmak için önleme yapılmıştır.
- **Boyut indirmek için öznitelik çıkarılması:** Bankaların tarafından bir müşteri bilgilerine kayıt etmek için, ortalama 30 öznitelik kullanılmaktadır. Ayrıca ABD ülkesinde bir bankanın ortalama günlük işlem sayısı 100 milyon üzerinde kayıt edilmiştir. Bu bilgilere göre çok sayıda veri olduğunu ve bunun üzerinde çalışmak, hesaplama ve gerçek-zamanlı çalışmayı daha da zorlaştırmaktadır. Bu yüzden, bir deneyim olarak, mevcut veri setin üzerinde otomatik kodlayıcı model geliştirilip bir öznitelik

çıkarma yapılmıştır. Böylece boyut indirme sağlayıp, zaman kazanmak adına, ikinci veri seti oluşturulmuştur.

- **Solo modellerin geliştirilmesi:** Kredi kartı dolandırıcılık alanında daha önce bütün yapılan araştırmaları incelemekten sonra, en iyi iki derin öğrenme modeli seçtik. Bunlardan birisi Kısıtlı Boltzmann Makinesi ve diğeri Kısa Uzun Süre Bellek'tır. RBM modelin yapısı bir işleme etiketleyip sınıflandırmaya çalışmak yerine, her zaman normal dağılımı tutmaktadır. Böylece gelen işlemin normal dağılımdan dışarıdaysa şüpheli olarak tanımlaya bilmekteyiz. Bu yöntem ile yenilikçi hileli işlemleri, başka deyişle daha önce bilinmeyen dolandırıcılık işlemleri tespit edebilmekteyiz. LSTM modelinde daha önce bilinen hileli işlemler hafızada tuttuğu için, çok iyi bir performansla hatırlayıp etiketleyebilmektedir.
- **Solo modellerin değerlendirilmesi:** Bahse geçen iki derin öğrenme modelin performansını diğer literatürde yer alan güncel araştırmalarla detaylı bir şekilde değerlendirilmiştir. Değerlendirme metriklerimiz geri çağırma, kesinlik, F1-ölçüt, AUC-ROC, AUC-PR ve süre olarak kayıt edilmiştir. Doğruluk metriği çok dengesiz bir veri setin üzerinde çalıştığımızdan dolayı hep yüksek bir miktar (%98) üzerinde olup, araştırmada anlamsız bulunup kullanılmamıştır.
- **Topluluk sınıflandırıcı modelin geliştirilmesi:** Kredi kartı dolandırıcılık probleminde hangi topluluk sınıflandırıcı modeli kullanmayı öğrenmek için, diğer son araştırmaları incelenmiş olup bagging topluluk sınıflandırıcı modelin kullanılmasına karar verilmiştir. Bu kararda önce hesaplamayı sonra performansı göz önünde bulundurmuş olup hız artırma amacımıza dikkat edilmiştir. Bagging topluluk sınıflandırıcı modeli geliştirmek için daha önce geliştirilip kayıt edilen RBM ve LSTM modelini çağırarak, %50 şekilde, yani eşit oranda, her bir işlem için iki modelin tahmini kullanılmıştır.
- **Karar destek sistemin değerlendirilmesi:** Geliştirilen karar destek sistemin performansını diğer araştırmalar ile bahse geçen

değerlendirme metriklerin üzerinde detaylı bir şekilde kıyaslanmıştır.

Doktora eğitim ve tez sürecinde farklı problemler için çeşitli çalışmalar gerçekleştirilmiştir. Bu çalışmalarda gerek yapay zeka yöntemleri test edilmiştir. Elde edilen başarılı sonuçlar bilimsel yayınlar ile desteklenmiştir. Gerçekleştirilen bu çalışmalardan elde edilen bilgilerden tez çalışmasında faydalanılmıştır. Bilimsel yayın haline getirilen çalışma submit aşamasında olup şu şekildedir:

1. Baghdadi, P., Korukoglu, S., Aytac, V., Onan, A., (2023). Implementation of a Decision Support System for Credit Card Fraud Detection Using Ensemble Deep Learning Models. *Cognitive Computation and Systems*, Wiley.

Bu çalışmada kredi kartı dolandırıcılık tespiti için bir karar destek sistemi gerçekleştirilmesi incelenmiştir. Elde edilen sonuçlar doğrultusunda gerçek zamanlı ve çok iyi bir performans sergileyen bir sistem geliştirilmesinin mümkün olduğu görülmüştür.

Tezin ikinci bölümünde literatürde yer alan kredi kartı dolandırıcılık tespiti çalışmalardan bahsedilmiştir. Kullanılmış olan veri setler, uygulanan ön işlemler, özniteliklerin çıkarılması ve boyut indirgemedede kullanılan öznitelik değerlendirme algoritmaları, otomatik kodlayıcı yöntemleri, üçüncü bölümde aktarılmıştır. Dördüncü bölümde derin öğrenme bazlı geliştirilen yöntemler ve ardından topluluk sınıflandırıcı modeller ve geliştirilmeleri aktarılmıştır. Beşinci bölümde elde edilen deneysel bulgular detaylı şekilde açıklanmıştır. Altıncı bölümde geliştirilen karar destek sistemin çalışma prensibi sunulmuştur. Yedinci bölümde edinilen bulgular tartışılmış ve diğer çalışmalar ile karşılaştırma sonuçlarına yer verilmiştir.

2. GEÇMİŞ ÇALIŞMALAR

Literatür incelendiğinde kredi kartı dolandırıcılık tespiti için az sayıda çeşitli verilerin kullanıldığı görülmüştür. Bu veriler şu şekilde listelenebilir:

- Avrupa kart sahipleri
- Brezilya kart sahipleri
- Avustralya kart sahipleri
- Almanya kart sahipleri
- Pysim uygulama üzerinde oluşan simüle veri seti
- IEEE-CIS kredi kartı dolandırıcılık veri seti

Ülke ismi ile başlayan veri setleri, o ülkeye ait olan banka tarafından kaydedilen işlemlerden oluşmaktadır. Pysim ile adlandırılan veri seti, bir mobil uygulama tarafından simüle edilmiş işlemlerdir ve IEEE-CIS ise, Vesta Corporation isimli bir şirketin e-ticaret işlemlerden oluşmaktadır. Kredi kartı dolandırıcılık tespiti için yapılan araştırmalarda çoğunlukla Avrupa kart sahipleri 2013 veri setini yaygın bir şekilde kullandıklarını söyleyebiliriz. Bunun sebeplerinden, yeterli sayıda işlemlerin olması ve gerçek bir banka tarafından kaydedilmiş olmasıdır. Genel olarak, araştırmacılar bu veri setinin yanında diğer bir veri seti seçip, sonuçları Avrupa kart sahipleri, daha iyi bir performans sergilediğini kanıtlamışlar. Nadir araştırmalar ise kendi özel veri setini bulup üzerinde çalışmalar yapmışlar, ama sonuçların düşük olduğunu ve diğer araştırmalar ile kıyaslayamadıklarını açıklamışlar. Tezin kapsamı ve yukarıdaki liste göz önüne alındığında literatür araştırmasının iki başlık altında incelenmesine karar verilmiştir. i) Sadece Avrupa kart sahipleri 2013 veriler ile kredi kartı dolandırıcılık tespiti, ii) Avrupa kart sahipleri veri seti üzerinde boyut indirgeme yapıp, ikinci veri seti oluşturup kredi kartı dolandırıcılık tespit edilmesi.

2.1 Makine öğrenme metodları ile KKDT

Jon TS Quah et al. (2008) araştırmalarında, gerçek zamanlı dolandırıcılık tespitine odaklanmakta ve olası dolandırıcılık vakalarını çözmek için harcama modellerini anlamaya yönelik yeni ve yenilikçi bir yaklaşım sunmaktadırlar. Dolandırıcılığın tespiti amacıyla müşteri davranışlarının şifresini çözmek, filtrelemek ve analiz etmek için kendi kendini organize eden (SOM) bir haritadan yararlanmışlar. Veri seti, 100'den fazla ülkede 200 milyondan fazla müşterisi olan önde gelen Singapur'daki bankanın şubesinden alınmış bir veri

setidir. Yazıda gerçek sürenin belirtilmesine rağmen, sadece özellikleri azaltmak için SOM kullanılmış ve sisteme girişten o işlem için dolandırıcılık olup olmadığına karar verene kadar geçen süreye değinmemişler. Suraj Pati et al. (2018) büyük miktarda veri sorunu için analitik bir çerçeve önermekte olup Alman Kart sahibi veri setini temel alan Lojistik Regresyon, Karar Ağacı ve Rastgele Ormanı kullanmışlar. Elde edilen bulgulara göre, yöntemlerin aralarında en iyi performansa sahip olan, Rastgele Orman olarak tespit etmişler. Nuno Carneiro et al. (2017) makalelerinde kullandıkları veri seti, 2016 yılında önde gelen çevrimiçi lüks moda perakendecilerinden biriyle ilgilidir. Lojistik Regresyon, Rastgele Orman ve SVM gibi gözetimli makine öğrenimi modellerini kullanmışlar. Tanımlanmış eşiklere sahip ML modellerine dayalı bir risk puanlama sistemi oluşturmuşlar. Naoufal Rtayli (2020a), büyük ve yüksek boyutlu verilerin sorununu çözmek amacıyla dolandırıcılık riskini tanımlamak adına özellik seçimine, Rastgele Ormana ve SVM sınıflandırıcısına dayalı gelişmiş bir CCRI önermiştir. Her işlem için dolandırıcılık riskini tespit etmek amacıyla Avrupa Kart sahibi veri kümesini kullanmış olup önerilen model, her işlem için özellik önem puanlarını hesaplamaktadır. Sara Makki (2020), finans sektöründeki sahtekarlıkları tespit etmek amacıyla çarpık verileri analiz etmek adına etkili bir sınıflandırma modeli önermiştir. Topluluk sınıflandırıcı olarak K-Ortalama (KNN tabanlı) dengesiz sınıflandırma üzerinde ve çarpık veri sorununu çözmeye çalışmış. Kendi araştırmasını LR, ANN, NB, CART ile karşılaştırmış ve sonuçları en iyi Sensitivity ve F1-ölçüt verirken, en iyi süre metriğinde, CART modelinden sonra en iyi sonucu göstermiştir. Naoufal Rtayli et al. (2020b), üç tekniğe dayalı hibrit model önermişler: En iyi tahmine dayalı özellikleri çıkarmak için, SVM-RFE tekniği, dengesiz bir veri kümesini dengeli bir veri kümesine dönüştürmek için SMOTE tekniği ve Kredi Kartı Dolandırıcılıklarını tespit etmek için Rastgele Orman Sınıflandırıcı kullanmışlardır. Veri kümeleri Avrupa Kart sahibi, Pysim ve üçüncü bankanın veri kümelerine kullanarak, verimlilik ve etkinlik açısından SVM, ANN, LR, NB ve KNN ile karşılaştırıp en iyi sonucu elde etmişlerdir. Akila Somasundaram et al. (2019), işlem penceresi torbalama yöntemini sunmuş. Bu yöntemin temel amacı artımlı öğrenmeyi dahil ederek kavram sapmasını ele almak ve paralelleştirilmiş bir torbalama topluluğu tabanlı dolandırıcılık tespit model oluşturmaktır. Araştırmada Brezilya Kartı sahibi ve UCSD veri seti kullanılmıştır. Modellerini AIRS, AIS tabanlı sahtekarlık tespit modeli ve Maliyete Duyarlı Sinir Ağı ile karşılaştırmalar ve her iki veri kümesinde de en yüksek AUC ve sahtekarlık tespit oranını ve en düşük maliyeti elde etmişlerdir. Alex G.C. de sa et al. (2018) kredi kartı sahtekarlığının tespiti için özelleştirilmiş NB sınıflandırıcısı sunmuşlar ve modelinin veri setini Brezilya'nın en popüler

çevrimiçi ödeme hizmeti olan PagSeguro'dan almışlardır. Sonuçları Naive Bayesian, Lojistik Regresyon, SVM, Rastgele Orman ile karşılaştırılmış ve modellerinin karışıklık matrisinde daha iyi performansa sahip olduğunu kanıtlamışlardır. Ashkan Zakryazad et al. (2016) modellerin, dolandırıcılık tespitinde, UCI deposundan alınan iki gerçek hayat veri seti üzerinde eğitmişlerdir. Yazarlar, doğrudan pazarlamada dolandırıcılık tespiti için, yanlış sınıflandırmayı maliyet olarak cezalandırırken doğru sınıflandırmanın kârını hesaplayan kâr odaklı bir yaklaşımla yapay sinir ağının farklı bir versiyonunu sunmuştur. Yusuf Şahin et al. (2013), terminal olmayan her düğümde bölme özelliğini seçerken yanlış sınıflandırma maliyetlerinin toplamını en aza indiren yeni bir maliyete duyarlı karar ağacı yaklaşımı sunmuş ve bu yaklaşımın performansı diğer makine öğrenme modeller ile karşılaştırılmıştır. Sonuçlar, bu maliyete duyarlı karar ağacı algoritmasının, doğruluk ve gerçek pozitif oran açısından verilen problem setinde mevcut iyi bilinen yöntemlerden daha iyi performans sunduğunu göstermektedir. Vaishnavi Nath Dornadula (2019), banka müşterilerinin davranış alışkanlıklarını çekmek için farklı makine öğrenme tabanlı sınıflandırıcılar kullanan kayan pencere stratejisini kullanarak kayan pencere kavramı problemini çözmeye çalışmıştır. Çalışmalarında Avrupa Kart sahibi veri setini kullanmakta olup performans değerlendirme metrikleri doğruluk, kesinlik ve MCC'dir. Araştırmada sonuçları SVM, Lojistik Regresyon, Karar Ağacı ve Rastgele Orman ile karşılaştırmışlar ve Lojistik regresyon, karar ağacı ve rastgele ormanın daha iyi sonuçlar veren algoritmalar olduğunu gözlemlemişlerdir.

Shahana T et al. (2023) dolandırıcılık tespitinde, kullanılan veri analitiği araçları, dolandırıcı firmaları tespit etmek için kullanılan veri tabanları, kontrol grubu örneklerinin tasarımı (dolandırıcı olmayan firmalar), kullanılan kritik boyut azaltma araçları, veri nadirliğini gidermek için benimsenen teknikler hakkında kısa bir genel bakış sunmaktadır. Araştırmaları dengesiz veriler, modelde kullanılan açıklayıcı değişkenler, dolandırıcılık göstergelerini destekleyen teorik çerçeve, kullanılan optimizasyon teknikleri, değerlendirme metriklerinin kullanımı ve önemli bulguları içermektedir. Rafael Van Belle et al. (2023) Temsil öğrenimine (RL) dayalı yeni bir ağ tabanlı kredi kartı dolandırıcılığına tespit yöntemi olan CATCHM'yi önermiştir. Yenilikçi ağ tasarımı, verimli bir tümevarımsal havuzlama operatörü ve dikkatli aşağı akış sınıflandırıcı yapılandırması aracılığıyla, manuel özellik mühendisliğinden kaçınarak ve işlemlerin ilişkisel yapısını açıkça dikkate alarak ağ RL'nin dolandırıcılık tespitinden nasıl yararlanabileceğini göstermiştir.

Asma Cherif et al. (2022) KKDT hakkında bir kapsamlı ve güncel araştırma sunmuşlar. İçerisinde büyük veri analitiği gibi yeni teknolojilerin kullanımı yoluyla kredi kartı dolandırıcılığın tespit edilmesiyle ilgili zorlukların ele alınması için daha fazla araştırmaya ihtiyaç olduğunu ortaya koymuşlar. Derin öğrenmeye, büyük ölçekli makine öğrenimi ve bulut bilişim yönelik bugüne kadar sınırlı araştırmayı gösteren bir inceleme önermiştir.

Palak Gupta et al. (2023) Veri dengeleme amacıyla kredi kartı dolandırıcılık tespitinde LR, DT, XGBoost ve ANN sınıflandırıcılarını karşılaştırmışlar. Bu nedenle modellerini hem aşırı örnekleme kullanarak, hem kullanmayarak veri setleri üzerinde denemişlerdir. Değerlendirme ölçümleri için kesinlik, geri çağırma, F1 ölçümü ve doğruluk kullanmışlar. Debjyoti Bagchi et al. (2021) kredi dolandırıcılığı, kredi kartı dolandırıcılığı ve şirket içi dolandırıcılıklarla ilgili çalışmışlar ve yanlış alarm oranı, gerçek pozitif oran, maliyet ve doğruluk ölçütlerine dayalı bir karşılaştırma çalışması olan ML ve torbalama topluluğu modelleri sunmuştur. Fabrizio Carcillo et al. (2019) gözetimli ve gözetimsiz teknikleri ile birleşen hibrit teknik sunarak, doğruluk metriği iyileştirmeye çalışmışlar. Salvatore Carta et al. (2019), e-ticaret işlemleri için bir yenilikçi veri zeka tekniği, PMCM olarak sunmuşlardır. Bu model, son teknoloji sınıflandırıcı algoritmalarının etkinliğini birleştirerek, olasılıksal ve çoğunluğa dayalı iki yönlü bir kriteri benimsemektedir. Amaç, herhangi bir veri dengesizliğinin varlığına bakılmaksızın, sahtekarlık işlemlerini tespit etmede modelin etkinliğini en üst düzeye çıkarmaktır. Vaishnavi Nath Dornadula et al. (2019) hazırlanmış oldukları çalışmada makine öğrenime algoritmalarını kullanarak kredi kartı dolandırıcılık tespiti gerçekleştirmişlerdir. Makalenin temel amacı, müşterilerin geçmiş işlem ayrıntılarını analiz edip, davranış kalıplarını çıkarmak amacıyla işlem verilerinin akışı için yeni bir dolandırıcılık tespit yöntemi tasarlayıp geliştirmişlerdir. Çalışmalarında, kart sahiplerinin işlem tutarlarına göre, farklı gruplara ayırt etmişlerdir. Daha sonra, farklı gruplardan kart sahipleri tarafından yapılan işlemleri bir araya getirmek için kayan pencere stratejisini kullanarak, grupların davranış modeli sırasıyla çıkarabilmişler ve en son, farklı sınıflandırıcıların gruplar üzerinde ayrı şekilde eğitmişlerdir. Araştırmanın sonucunda Lojistik Regresyon, Karar Ağacı ve Rastgele Ormanın daha iyi algoritmalar olduğunu gözlemlenmiştir. Sangeeta Mittal et al. (2019) hazırladıkları çalışmalarında, kredi kartı dolandırıcılık tespiti için makine öğrenme algoritmalarının performansları kıyaslanmıştır. Bu çalışmada popüler gözetimli ve gözetimsiz makine öğrenme algoritmaları uygulanmış olup metrik açısından, kesinlik, negatif tahmin değeri, duyarlılık, dengelik doğruluk, yaygınlık, teşhis odd oranı dikkate alınmıştır. Sonuçlardan,

sınıflandırma mekanizmalarından hiçbirinin her bakımdan en iyi olmadığı görülmüştür. Bu nedenle, sonuçlar FDS kullanıcıları ile ilgili performans sorularına göre incelenmiş ve yorumlanmış. Ayrıca, Gözetimsiz algoritmaların dengesiz veri kümesinde daha iyi performans gösterdiğini yorumlamışlardır. Jon Ander Gomez et al. (2017) kart ödemelerinde dolandırıcılık puanlaması için sinir ağı tabanlı bir sistem sunmuşlardır. Gerçek dünya veri kümesini kullanıp dengesizlik, veri işleme ve maliyet ölçütü değerlendirmesi gibi önemli hususlara odaklanarak, uygulayıcının bakış açısından bir çözüm tanımlamışlardır. Çalışmalarının amacı, özellikle dengesizliği ve maliyet metrik değerlendirmesini azaltmak için bir dizi sınıflandırıcı kullanıp, yapay sinir ağları kullanarak bu sorunun nasıl çözüleceği konusunda veri uygulayıcılarına bir fikir vermektir. Bu makalenin deneysel kısmı, önerilen yaklaşımı kullanarak gerçek verilerle en gelişmiş performansı elde etmenin mümkün olduğunu göstermiştir. Akila Somasundaram et al. (2018) araştırmalarında, kredi kartı işlem verilerindeki sorunları ele almak için TWB adına bir paralel ve artımlı öğrenme topluluğu modeli önermişlerdir. Bunu yürütmek için Brezilya Bankası verileri ve UCSD verileri ile deneyler yapıp sonuçları en son modellerle karşılaştırmışlardır. Önerilen model, gerçek zamanlı olarak etkin bir şekilde kullanılabilecek düşük gecikmeli bir modele yol açmış. Fabrizio Carcillo et al. (2019) kredi kartı dolandırıcılık tespitinde gözetimli ve gözetimsiz öğrenme model birleştiren bir hibrit teknik sunmuşlar. Tespitin kalitesini ölçmek için literatürde AUC-ROC, AUC-PR, F1-ölçüt, özgüllük, geri çağırma ve kesinlik dahil olmak üzere çeşitli metrikler önermişlerdir. Katkının yeniliği, kredi kartı işlemlerinin gerçek ve büyük veri kümelerindeki uygulamalarının ötesinde, aykırı değer puanının tanımı için farklı ayrıntı düzeylerinin uygulanıp değerlendirilmiştir.

Waleed Hilal et al. (2022) yarı denetimli ve denetimsiz öğrenme alanlarındaki son gelişmeleri vurgulamaya odaklanarak, mali dolandırıcılığı tespit etmek için uygulanan en popüler ve etkili anormallik tespit tekniklerini araştırmış olup kapsamlı bir incelemeyi sunmayı amaçlamışlardır. Siddhant Bagga et al. (2020) Pipeling'in çapraz doğrulama sağladığı, aynı anda farklı parametreleri ayarladığı ve sınıflandırıcıyı çeşitli sınıflandırıcıların birleşimi için bir topluluk modeli olarak paketlediği bir Pipeling topluluk öğrenme modeli önermişler. Yazarlar topluluk tabanlı sınıflandırıcısından bahsetmemekte ve karar ağacına benzeyen “kara kutu” terimini kullanmışlar. Veri setlerini ECH iken sonuçlarını LR, KNN, RF, NB, MLP, ada boost ile karşılaştırmışlardır.

J. Femila Roseline et al. (2022) çalışmalarının amacı kredi kartı dolandırıcılığının nasıl tespit edileceğini bulmaktır. Dolandırıcılık olayını tespit

etmek için Uzun Kısa Süreli Bellek - Tekrarlayan Sinir Ağı (LSTM-RNN) önerilmektedir. Ayrıca performansı daha da artırmak için bir dikkat mekanizması dahil etmişlerdir. ECH veri setini kullanmışlar ve çalışma için iki dağılım seti elde etmek amacıyla dengesiz veri seti üzerinde Aşırı Örnekleme gerçekleştirmişler. Hassasiyet, geri çağırma, f1-ölçümü ve modellerinde 100% sonuç elde ettiklerini kayıt etmişler ancak aşırı uyum sorunu varmış gibi görünmektedir. Jonathan Kwaku Afriyie et al. (2023) Sahte kredi kartı işlemlerini sınıflandırmak, tahmin etmek ve tespit etmek için lojistik regresyon, rastgele orman ve karar ağaçları olmak üzere üç farklı makine öğrenimi modelinin performansını incelemişler ve bu modellerin performansını karşılaştırarak, rastgele orman'ın, maksimum %96'lık bir doğruluk ürettiğini kanıtlamışlar ve aralarında en iyi model olduğunu belirtmişlerdir.

2.2 Derin öğrenme metodları ile KKDT

Javad Forough et al. (2020) çalışmalarında Kredi kartı dolandırıcılığı tespiti için derin sıralı modeller topluluğu kullanarak bir yeni model elde etmişlerdir. Bu makalede, derin tekrarlayan sinir ağları kullanılarak verilerin sıralı modellenmesine dayalı bir topluluk modeli ve sahte eylemleri tespit etmek için yapay sinir ağına dayalı yeni bir oylama mekanizması önermişlerdir. değerlendirme metrikleri geri çağırma, kesinlik, F1-ölçüt, AUC-ROC ve AUC-PR içerir ve değerlendirme kriterlerini karşılaştırmak için iki gerçek dünya veri seti üzerinde, yani Avrupa ve Brezilya kart sahipleri veri setlerin üzerinde çalışmışlardır. Topluluk modelleri için iki tür derin öğrenme tabanlı; bir dizi LSTM veya bir dizi GRU ağlarını temel sınıflandırıcılar olarak kullanmışlar. Bir sonraki adımda, oylama mekanizması olarak bir İleri Beslemeli Sinir Ağı kullanarak, tüm temel sınıflandırıcıların sonuçlarını, mevcut işlemin tahmin edilen sınıfı olarak kabul edilecek tek bir ikili değer üreten bir sinir ağına toplamışlar. Test edilen modellerin süre analizi, LSTM ve GRU tabanlı topluluk modellerindeki eğitim ve test süresinin miktarının, temel sınıflandırıcıların sayısı ile doğrudan bir ilişkisi olduğunu kanıtlamışlar. Ayrıca, GRU tabanlı topluluk modeli, eğitim ve test süresi açısından LSTM tabanlı olana göre daha iyi verimliliğe sahip olduğunu belirlemişler. Youness Abakarim et al. (2018) derin öğrenmeye dayalı Kredi Kartı Dolandırıcılık tespiti için etkili bir gerçek zamanlı model üzerinde çalışmışlar. Önerilen modelleri bir otomatik kodlayıcıya dayalıdır ve gerçek zamanlı olarak kredi kartı işlemlerini gerçek veya sahte olarak sınıflandırmaya izin verdiği belirtilmiştir. Modellerinin etkinliğini test etmek için karşılaştırma olarak dört farklı binary sınıflandırma modeli; Linear-SVM, Lojistik Regresyon, Non-linear Auto-Regression ve Otomatik Kodlayıcı

ile derin yapay sinir olarak kullanmışlar. Ayrıca sadece kesinlik, geri çağırma ve F1-ölçüt metrik olarak düşünmüşler. Modellerin arasında, Otomatik Kodlayıcı, genel olarak en iyi F1-ölçüte sahip olduğunu ve ardından Lojistik Regresyon olduğunu kanıtlamışlardır. Xinwei Zhang et al. (2019) derin öğrenme mimarisine dayalı kredi kartı dolandırıcılığı tespiti için yeni bir özellik mühendisliği metodolojisini geliştirmişler. Çalışmalarının ana katkısı, homojenlik odaklı davranış analizine dayalı gelişmiş bir özellik mühendisliği süreci ile birlikte derin bir öğrenme mimarisi kullanan bir dolandırıcılık tespit sisteminin geliştirilmesidir. Bu çalışmada, Çin'deki en büyük ticari bankalardan birinden gerçek hayattaki bir veri kümesini kullanmışlar. Sonuçları, derin öğrenme yaklaşımlarının, RFM özellik seti veya HODA özellik seti ne olursa olsun, diğer tüm temel çizgilerden önemli ölçüde daha iyi performans gösterdiğini kanıtlamışlardır. HODA değişkenlerine sahip Derin İnanç Ağı sınıflandırıcısı, kıyaslama modellerinden önemli ölçüde daha iyi bir performans elde etmesini ibraz etmişlerdir. Sumit Misra et al. (2019) kredi kartı dolandırıcılık işlemin tespiti için Otomatik Kodlayıcı bazlı bir model geliştirmişler. Bu çalışmada, kredi kartı işlemlerinde sahte olanları tespit etmek için iki aşamalı bir model önermişlerdir. Önerilen modelin ilk aşamasında otomatik kodlayıcı, işlem özniteliklerini daha düşük boyutlu bir özellik vektörüne dönüştürerek bu yöne odaklanmışlardır. Bu tür öznitelik vektörleri daha sonra ikinci aşamada bir sınıflandırıcıda kullanılmıştır. Deneysel sonuçlar, önerilen metodolojinin hileleri tespit etmede kesinlik ve geri çağırma arasında iyi bir denge sağladığını göstermişlerdir. Ayrıca, otomatik kodlayıcının farklı sınıflandırıcılara veya türevlerine dayalı sistemlerde daha iyi performans gösterdiğini açıklamışlardır. Apapan Pumsirirat et al. (2018) kredi kartı dolandırıcılık tespiti için, derin öğrenme bazlı Otomatik Kodlayıcı ve Kısıtlı Boltzmann Makinesine dayalı bir model tasarlamışlardır. Bu makalede, 1) önceki geçmişe veya gözetimli öğrenmeye dayalı olarak tespit edilemeyen dolandırıcılık vakalarına odaklanmayı, 2) normal kalıplardan anormallikleri bulmak için normal işlemleri yeniden yapılandırabilen bir derin Otomatik kodlayıcı ve Kısıtlı Boltzmann Makinesi modeli oluşturmayı amaçlamışlardır. Bu kapsamda 3 ünlü veri setlerden, yani Avrupa, Avustralya ve Almanya kart sahipleri veri setlerini kullanmışlardır. Üç veri setin arasında, Almanya ve Avustralya veri setlerinde daha az veri bulunmaktadır ve bu nedenle, dolandırıcılık tespitinde anormallikler bulduklarında, sistemleri az sayıda veri için eğittikleri ve test verilerinin daha az miktarda doğruladıkları için daha düşük bir AUC elde etmişler. Tersine, Keras'a dayalı bu AE modelini büyük miktarda, Avrupa kart sahipleri veri kümesi ile uygulamalarında yüksek miktar AUC elde etmişlerdir. AE, büyük veri kümeleri için uygun olduğunu ve Avrupa veri setinin RBM'deki AUC puanı, AE'den biraz

daha düşük olduğunu göstermişler. Bu araştırmada, AE ve RBM'nin daha büyük veri kümeleri için yüksek AUC puanları ve doğruluk ürettiği sonucuna ulaşmışlardır.

Yu-Tian Lei et al. (2023) Yapay sinir ağların üzerinde bir dağıtılmış ağ oluşturmuşlar. ECH veri seti kullanarak, eğitim setlerine 4 bölüme dağıtıyorlar. Test setini dağıtırken ve merkezi olanla farklı dönemlerdeki kayıp değerini karşılaştırmışlardır. Sonuçlar, üzerinde çalışılan modeller arasında etkileyici farklılıklar göstermese de araştırmacılar, dağıtılmış veri kümesi maliyet yönetimi ve gizlilik sızıntısının avantajlarından bahsedilmiştir. Bunun yanı sıra deneysel sonuçlar, bu çalışmada önerilen DDNN'nin merkezi olarak oluşturulan modellere göre modeli daha hızlı optimize ettiğini ve daha sağlam tespit yeteneğine sahip olduğunu göstermektedir. Hosein Fanai et al (2023) çalışmalarında, sahte işlemleri tespit etmek için temsili öğrenme yöntemi olarak derin AE'yi ve denetimli derin öğrenme tekniklerini içeren iki aşamalı bir çerçeve önermişler. ECH ve Alman veri kümelerinde AE'yi kullanarak, özellik çıkarımı kullanmışlar ve bunu PCA özellik çıkarımı ile karşılaştırmışlardır. Temel sınıflandırıcıları olarak CNN-RNN, DNN, RNN ve SVM kullanırken, AE tabanlı özellik çıkarımının PCA ile karşılaştırıldığında üstünlüğü kanıtlamışlar. Banghua Wu et al. (2023) araştırmalarında veri kümelerin üzerinde, over-sampling, under-sampling ve yeniden örnekleme yapmadan, temel sınıflandırıcı olarak makine öğrenme ve derin öğrenmeyi kullanmışlardır. Veri kümeleri 7,352,421 işlem içeren Özel Master Kart olarak kullanılmış ve doğrusal regresyon, rastgele orman ve LSTM'yi karşılaştırmışlardır, ancak sonuçları etkileyici olmamış olup diğer araştırmalara göre düşük performans sergilemiştir.

Karthikeyan, T et al. (2023) Araştırmalarının hedefleri, önceki kayıtlar veya denetimli öğrenme (CSO) kullanılarak tespit edilemeyen dolandırıcılık senaryolarına özellikle vurgu yaparak, rekabetçi sürü optimizasyonu tarafından üretilen düzenli modellerdeki anormallikleri tespit etmek için derin bir evrişimli sinir ağı modeli oluşturmaktır. Önerilen CSO-DCNN adı verilen denetimsiz bir öğrenme yöntemi, girdileri ve çıktıları eşitleyerek ReLu'yu kullanmışlar. Değerlendirme ölçütleri Doğruluk, Ortalama kare hata ve Ortalama Mutlak Hata'yi kullanmışlardır. Zhang Runchi et al. (2023) Lojistik-BWE modeli olarak lojistik regresyona dayalı yeni bir topluluk modeli tasarlamışlar. İlk önce veri ön işlemeyi gerçekleştirmişlerdir, ardından farklı dengesizlik oranlarına sahip çeşitli eğitim alt veri kümeleri oluşturmak ve sırasıyla alt modeller oluşturmak için bir örnek dengeleme algoritması uygulamış olup son olarak doğrulama aşamasında her bir alt modelin performansına göre, tahmin edilenin ağırlığı Her

alt modelin farklı sınıflarına ait sonuçlar dinamik olarak hesaplamıştır. On farklı sınıflandırıcı ve birden fazla veri seti arasında karşılaştırma yapan kapsamlı bir makale olduğunu söyleyebiliriz.

Md Amirul Islam et al. (2023) Kredi kartı anormalliklerinin tespit oranını iyileştirmek için temel öğrenen paradigmasını ve meta-öğrenme topluluğu tekniklerini kullanan Kredi Kartı Anomalisi Tespiti (CCAD) adı verilen bir model önermişler. Kredi kartı işlemlerinde anormallikleri tespit etmek için önerilen yığılmış topluluk yaklaşımında temel öğrenciler olarak dört aykırı değer tespit algoritması ve meta öğrenci olarak XGBoost algoritmasını kullanmışlar. Veri dengesizliği ve aşırı uyum sorunlarını çözmek için katmanlı bir örnekleme tekniği ve k-katlı çapraz doğrulama süreci uygulamışlardır. Ayrıca topluluk öğrenme performanslarının doğruluğunu artırmak için uyumsuzluk oranı da hesaplanmış. Önerilen model iki veri kümesi kullanılarak eğitilmiş ve test edilmiştir: CCF (Kredi Kartı Dolandırıcılığı) ve CCDP (Kredi Kartı Varsayılan Ödeme).

Seyyede Zahra Aftabi et al. (2023) üretken çekişmeli ağlara (GAN) ve topluluk modellerine dayalı, yalnızca sahte olmayan örneklerin eksikliğini gidermekle kalmayıp aynı zamanda özellik alanının yüksek boyutluluğunu da ele alabilen yeni bir yaklaşım önermişlerdir. Ayrıca, on İran bankasının 2014 - 2019 yılları arasındaki yıllık mali tablolarının toplanması ve ardından bu çalışmada önerilen üç tür özelliğin çıkarılmasıyla yeni bir veri seti oluşturmuşlardır. Veri setlerini 4 parçaya ayırmışlar: Krediler, Gelirler, Mevduatlar ve Maliyetler. Dolandırıcılığa yatkın örnekleri doğru bir şekilde ayırt etmede denetlenen modellerle karşılaştırmalı performansa ve denetlenmeyen modellere göre daha iyi performansa ulaştığının sonucuna varmışlardır. Baio Xu et al. (2023) 5 farklı veri seti üzerinde çalışmışlar ve bunlardan birisi ünlü ECH'miş. Tüm dolandırıcılık türlerinde en iyi performansı aşan genel bir makine öğrenimi algoritması bulmayı hedeflemişlerdir. DBDT olarak Derin Güçlendirme Karar Ağacı, önerilen modeldir ve temsili öğrenme yeteneğini geliştirmek ve yorumlanabilirliği korumak için sinir ağlarını gradyan artırmaya çalışmışlardır. Ayrıca, tespit edilen dolandırıcılık vakalarının nadirliğini hedefleyerek, model eğitimi aşamasında, algoritma düzeyindeki veri dengesizlikleriyle başa çıkmak için bileşimsel bir AUC maksimizasyon yaklaşımı önermişlerdir.

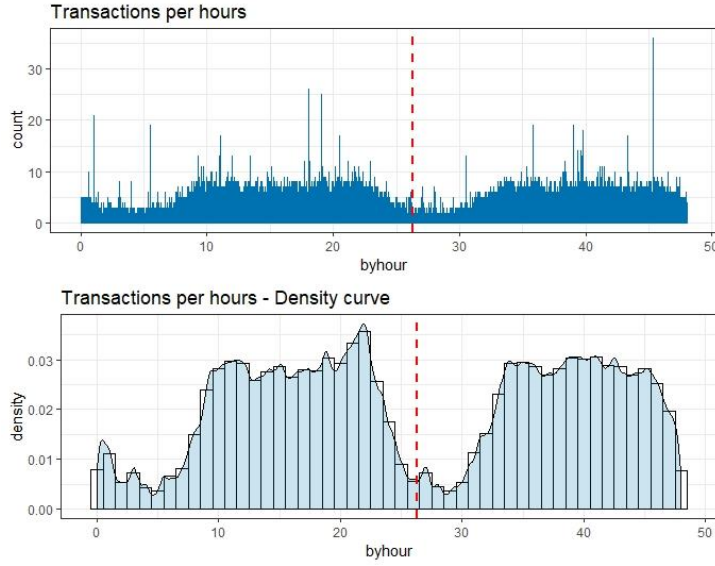
3. VERİ KÜMESİ

Bu tez çalışmasında, bankaların ve finans kuruluşlarının işlemlere ilişkin dolandırıcılık veri setlerini ortaya çıkarma konusundaki isteksizliklerine dikkat edilerek (Cherif et al. 2022), kullanılabilecek sınırlı sayıda kredi kartı dolandırıcılık tespiti için, gerçek banka veya finansal kurumların veri seti üzerinde yani Avrupa Kart sahipleri 2013 veri seti kullanılmıştır. Bu veri seti kredi kartı dolandırıcılığı araştırmalarda en yaygın olarak kullanılan veri setlerinden birisidir.

3.1 Veri Analizi

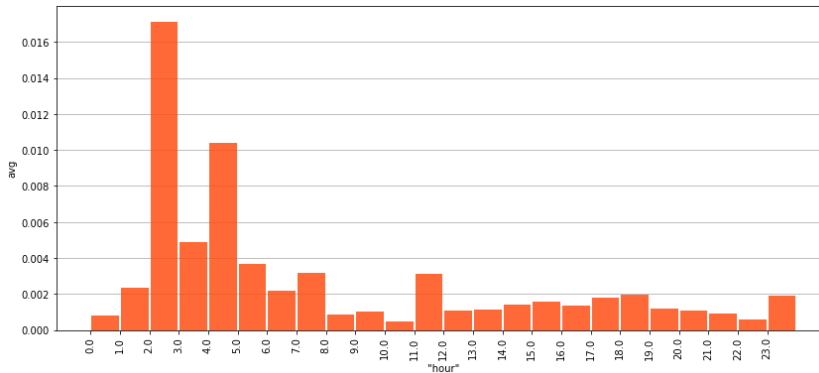
Avrupa kart sahipleri 2013 veri kümesinde toplam 284.807 işlem bulunmakta olup 492 dolandırıcılığın olduğu iki gün içinde gerçekleşen işlemleri sergilemektedir. Bu veri setinin zorlu yanı, oldukça dengesiz olması ve pozitif sınıfın (dolandırıcılık) tüm işlemlerin %0,172'sini oluşturmasıdır. Yukarıda bahse geçen mahremiyet ve güvenlik sebeplerden dolayı, veri setinin üzerinde PCA dönüşümü yapılmış ve sadece sayısal girdi değişkeni içermiştir. Bu nedenle, veri kümesindeki V1, V2, V3, ... V28 sütunları, PCA tarafından sağlanan temel bileşenlerdir. Bu veri setindeki dönüştürülmemiş özellikler “Süre”, “Miktar” ve “Sınıf”tır. “Süre” özelliği, her bir işlemin gerçekleşmesi ile ilk işlem arasındaki süreyi belirtir. “Tutar” işlemin parasal değeridir. Ayrıca, “Sınıf” etiketi, bir işlemin sahte olup olmadığını sırasıyla "1" ve "0" ikili değerleriyle göstermektedir.

Veri setimiz gerçekleşme anına göre sıralanmış ve ilk işlem 00:00 saatinde gerçekleşmiş olarak varsayıp, son işlem, birinci işlemten ne kadar sonra kayıt edildiğini göstermektedir. Şekil 3.1'de, yukarıdaki grafik, işlem sayısını saat bazında gösterirken aşağıdaki grafik, işlem dağılımını süre olarak göstermektedir.



Şekil 3.1. Kırk Sekiz saat içerisinde işlem dağılımı.

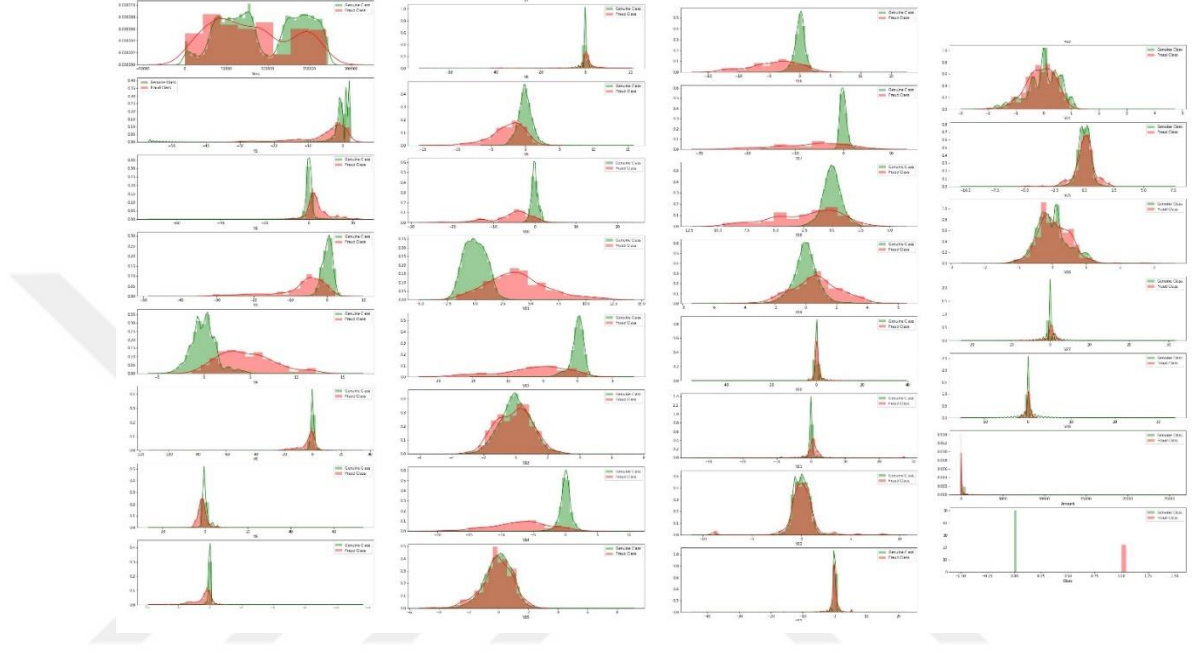
Şekil 3.1’de gösterilen dağılıma göre, gece saatlerdeki işlem sayılarının en az seviyede olduğunu ve saat 09:00 ile 22:00 arasında en yüksek seviyede olduğunu görebilmekteyiz. Ancak, dolandırıcılık işlemlerin dağılımını şekil 3.2’de gördüğümüzde, en yüksek dolandırıcılık işlemler saat 02:00 ile 05:00 arasında olduğunu fark edebiliriz. Bu resimlere göre, işlemlerin sayısı gece saatlerde minimum olsa bile dolandırıcılık davranışını maksimum olduğunu kanıtlayabiliriz.



Şekil 3.2. Yirmi dört saat içerisinde dolandırıcılık işlemlerin sıklığı.

Daha öncede belirttiğimiz gibi, V1 ile V28 sütunlarımız PCA tarafından dönüştürülmüş olup sayısal bir rakam olarak tanımlanmıştır ve bunların hakkında hiçbir ek bilğimiz yoktur. Ancak, bunları analiz etmek adına gerçek ve dolandırıcılık işlemler dağılımı Şekil 3.3’te sunulmuştur. Bu resimde, yeşil renkler normal işlemler olarak belirlenmişken, kırmızı renkler, dolandırıcılık işlemler olarak tanımlanmıştır. Bu şekilde, V13, V15, V20, V22, V23, V24,

V25, V26 sütunlarımızda işlem dağılımlar birbirine benzerken diğer sütunlarda güzel şekilde ayırt edildiğini anlayabiliriz. Bu da şu demek ki, bahse geçen sütunların model eğitimiz için şaşırtıcı olabilir, bu yüzden bazı araştırmalarda bunları görmezden gelmekte ve modellerinin performansını artmış olmaktadır, halbuki bazı önemli bilgilerin kayıp olmasına yol açabiliyor. Bu araştırmada bütün verileri tutulup kullanılmıştır.



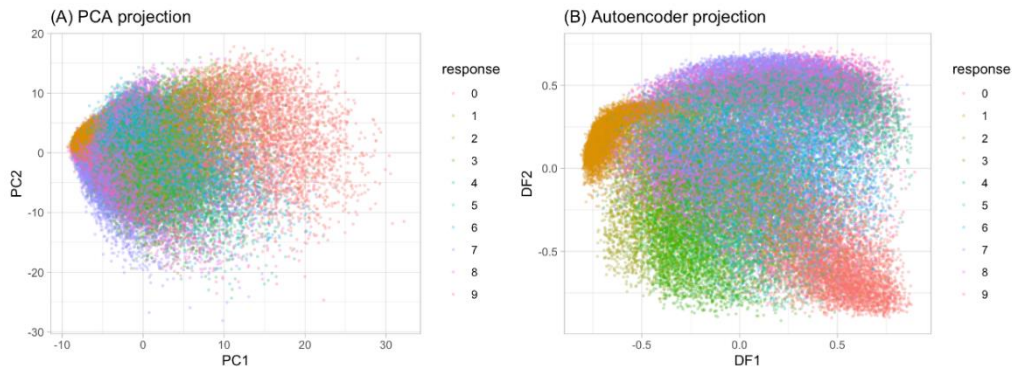
Şekil 3.3. Her özelliğe göre gerçek ve dolandırıcılık işlem dağılımı.

3.2 Veri setinde ön işlem yapılması

Kredi kartı dolandırıcılık alanındaki çoğu araştırmaları inceledikten sonra, Avrupa Kart Sahipleri 2013 veri seti, hem yaygın şekilde kullanılmış olup, diğer araştırmalar ile kıyaslamak için ideal olduğunu, hem de yeterli sayıda işlem var olduğundan dolayı, model eğitimiz için çok iyi bir örnek olduğunu söyleyebiliriz. Bu yüzden kendi araştırmamızda bu veri setini de kullanmış olduk, ancak diğer araştırmalarda uygun olmayan diğer veri setleri kullanırken, aynı veri seti özetleştirerek, 2. veri seti olarak kullanmaya karar verilmiştir. Özetleştirmekten amacımız öznitelik çıkarmak olup, boyut indirmiş olurken, gerçek zamanlı dolandırıcılığı tespit etmek adına hızı da arttırmış oluyoruz.

3.2.1 Veri setin öznitelik çıkarımı

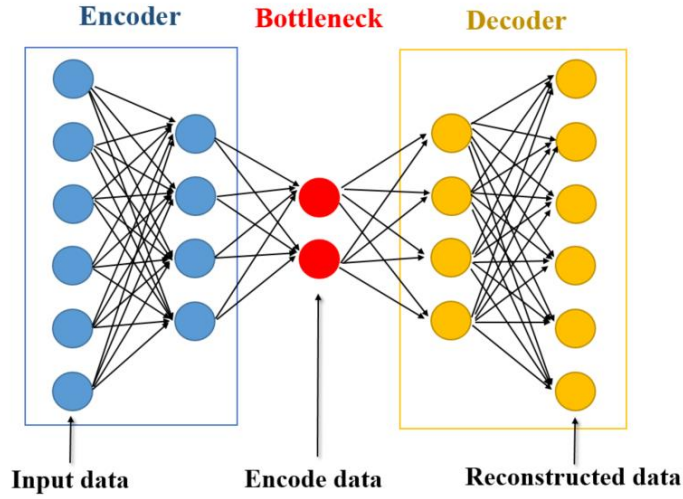
Gerçek dünyadaki makine öğrenme problemlerinde, genellikle son tahmin yapıldığında çok fazla faktör veya başka bir deyişle özellik vardır. Özellik sayısı ne kadar yüksek olursa, eğitim setini görselleştirmek ve üzerinde çalışmak o kadar zorlaşır. Bu özelliklerin bazıları çok önemli olup bazıları gereksiz olabilir, bu durumlarda boyut azaltma algoritmalar devreye girerler. Öte yandan, Özellik Çıkarma, bir veri kümesindeki özelliklerin sayısını, mevcut özelliklerden yeni özellikler oluşturarak ve ardından orijinal özellikleri atarak, azaltmayı amaçlar. Bu yeni azaltılmış özellik kümesi, orijinal özellik kümesinde yer alan bilgilerin çoğunu özetleyebilmelidir. Bu şekilde, orijinal setin bir kombinasyonundan orijinal özelliklerin özetlenmiş bir versiyonu oluşturulabilir. Öznitelik çıkarmak için en iyi yöntemler: Otomatik Kodlayıcı ve PCA dır. Otomatik Kodlayıcı, hem doğrusal olan hem de doğrusal olmayan yüzeyler için çalışırken, PCA yalnızca doğrusal yüzeylerde çalışıyor, halbuki gerçek dünya problemlerinde doğrusal olmayan problemler ile karşı karşıyız. Bir yandan, Otomatik Kodlayıcı yalnızca doğrusal etkinleştirme işlevlerini kullanmış olup kayıp işlevi MSE olduğunda, Otomatik Kodlayıcının PCA'ya dönüştüğünü görebiliyoruz. Şekil 3.4'te PCA ve Otomatik Kodlayıcı'nın arasındaki karşılaştırmayı göstermekteyiz. Şekilde görüldüğü gibi, sol tarafta PCA, doğrusal bir projeksiyonu zorlarken, sağ tarafta doğrusal olmayan etkinleştirme işlevlerine sahip bir otomatik kodlayıcı, doğrusal olmayan projeye izin vermiş olup daha anlamlı ayırt etmiştir.



Şekil 3.4. PCA ve Otomatik kodlayıcı tarafından doğrusal olmayan projeksiyonları.

3.2.2 Otomatik Kodlayıcı

Bir tipik Otomatik Kodlayıcı, girdi katmanında ne beslenirse, gizli katmanda o bilgileri özetleyerek, çıktı katmanda ona benzer bilgiyi üretmeye çalışmaktadır (Şekil 3.5). Aslında girdi katman (encoder) ve çıktı katman (decoder) arasında bir dizi az nöron içeren darboğaz katmanları (bottleneck) vardır ve girdideki kabul eden verileri, gizli katmanda önemli bilgilerini çıkarıp özetlerken, çıktı katmanda bu özetlenmiş bilgilerden yararlanarak, girdi katmandaki veriye benzer veri üretmeye çalışmaktadır (Machine learning mastery, 2022).



Şekil 3.5. Otomatik Kodlayıcının genel yapısı.

Otomatik Kodlayıcıların kullanılmasının ardındaki mantık, gizli katmanların girdiyi bir vektör uzayına eşlemesidir. Birkaç nöronlu bir katmanın sonucunu kullanarak, girdimizi yüksek boyutlu bir uzaydan daha az boyutlu bir alana eşlemekte oluyor. Gözetimsiz bir öğrenme yöntemidir, ancak teknik olarak kendi kendine-denetimli olarak adlandırılan gözetimli öğrenme yöntemleri kullanılarak eğitilirler.

Otomatik Kodlayıcılar genellikle girdiyi yeniden oluşturmaya çalışan daha geniş bir modelin parçası olarak eğitilir. Örneğin:

$$X = model.tahmin(X).$$

Otomatik Kodlayıcı modelinin tasarımı, mimariyi, giriş verilerinin yeniden yapılandırılmasının gerçekleştirildiği modelin orta noktasındaki bir bottleneck ile sınırlayarak, kasıtlı olarak bunu zorlaştırır.

3.2.3 Öznitelik Çıkarma İçin Otomatik Kodlayıcı Geliştirme

Otomatik Kodlayıcı modelin yapısını inceledikten sonra, Python yazılım dili ile öznitelik çıkarmak amacıyla, geliştirme yapılmıştır. İlk adımda veri seti okunduktan sonra, train verileri için veri setinin %80 (227,845) ve test verileri için %20 (56,962) ayırt edilmiştir. Ardından, girdi verileri ayarlamak için MinMaxScaler kullanarak, değerler 0 ile 1 arasında normalleştirilmiştir.

Otomatik Kodlayıcı Algoritması

Girdi: Her bir işlem için sınıf değeri ve öznitelik değerlerini içeren vektör. (X^t)

Çıktı: Eğitilmiş otomatik kodlayıcı model.

Kodlayıcı: Sigmoid aktivasyon fonksiyonuyla üç kodlayıcı katmanını tanımlayın.

1. kodlayıcı katmanı, 48 üniteli
2. kodlayıcı katmanı, 40 üniteli
3. kodlayıcı katmanı, 32 üniteli

Darboğaz: 16 üniteli ve sigmoid aktivasyon fonksiyonlu darboğaz katmanı.

Kod çözücü: Sigmoid aktivasyon fonksiyonuyla üç kod çözücü katmanını tanımlayın.

1. kod çözücü katmanı, 32 üniteli
2. kod çözücü katmanı, 40 üniteli
3. kod çözücü katmanı, 48 üniteli

1. **For** each training example X^t
 2. Give the input to the model
 3. Calculate the predicted input based on MSE
 4. Update parameters
 5. **Go to 1 until the last transaction**
-

Yukarıdaki getirilen sözde koddaki, kodlayıcı bölümü üç katmana sahip olacak şekilde tanımlanmıştır. Birinci katmanın giriş sayısı 48, ikinci ise 40 ve üçüncü 32 olarak tanımlanmıştır. Giriş verilerimiz 0 ile 1 aralığına standartlaştırdığımız için ve ayrıca problemimiz ikili sınıflandırma olduğundan dolayı, Sigmoid aktivasyon fonksiyonu kullanılmıştır. Darboğaz için giriş sayısı 16 olmuş olup Sigmoid aktivasyon fonksiyonu kullanılmıştır. Kod çözücü kısmı için Kodlayıcı kısmı ile aynı ayna konfigürasyon verilmiştir. Şekil 3.6'da yukarıdaki açıklamaların Python kodunun geliştirilmesi göstermektedir.

```

55 #define encoder
56 visible = Input(shape=(n_inputs,))
57
58 e = Dense(48, activation='sigmoid')(visible)#activation='relu')(visible)
59 #e = ReLU()(e)
60 e = Dense(40, activation='sigmoid')(e)
61 #e = ReLU()(e)
62 e = Dense(32, activation='sigmoid')(e)
63 #e = ReLU()(e)
64
65 # bottleneck
66 n_bottleneck = 16 #round(float(n_inputs) / 2.0) #n_inputs/2
67 bottleneck = Dense(n_bottleneck, activation='sigmoid')(e)#activation='relu')(e)
68
69 d = Dense(32, activation='sigmoid')(bottleneck)
70 #d = ReLU()(d)
71 d = Dense(40, activation='sigmoid')(d)
72 #d = ReLU()(d)
73 d = Dense(48, activation='sigmoid')(d)
74 #d = ReLU()(d)
75
76 # output layer
77 output = Dense(n_inputs, activation='sigmoid')(d)#activation='relu')(d)
78
79 # define autoencoder model
80 model = Model(inputs=visible, outputs=output)
81 # compile autoencoder model
82 model.compile(optimizer='adam', loss='mse')
83 # fit the autoencoder

```

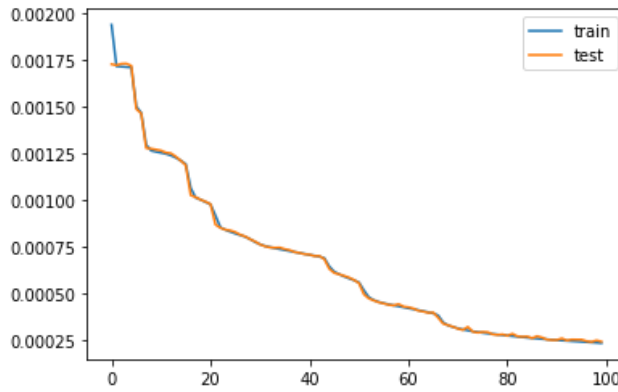
Şekil 3.6. Otomatik Kodlayıcının Python ile geliştirilmesi.

Model’de, stokastik gradyan inişinin etkin, Adam versiyonu kullanılarak derlenmiştir ve MSE’yi en aza seviyeye indirilmiştir. Konfigürasyon denemelerin sonucunda, 100 epoch ve 32’lik batch size kullanıldığında, modelin en iyi şekilde fit olduğu kayıt edilmiştir. Daha sonra, modeli, girdiyi yeniden üretecek şekilde eğitilmiştir ve bekletme test setinde modelin performansını takip edilmiştir. İlk ve son 3 epoch’un Kayıp ve Validation puanını 3.1. çizelgede getirilmiştir. Gördüğümüz gibi Loss ilk dönemden son döneme kadar 0,0049’dan 2.3061e-04’e düşmüştür. Ayrıca Validation hatası aynı dönemlerde 0,0045’ten 2.3882e-04’e düşürülmüştür.

Çizelge 3.1. 227845 numune üzerinde eğitilmiş sonuçlar.

Epoch	Loss
1/100	loss: 0.0049 - val_loss: 0.0045
2/100	loss: 0.0044 - val_loss: 0.0021
3/100	loss: 0.0020 - val_loss: 0.0020
98/100	loss: 2.3434e-04 - val_loss: 2.3729e-04
99/100	loss: 2.3288e-04 - val_loss: 2.4500e-04
100/100	loss: 2.3061e-04 - val_loss: 2.3882e-04

Şekil 3.7’deki öğrenme eğrilerinin bir grafiğini gösterip, modelin, fazla uydurma değil, belki eğitim boyunca sabit kalan girdiyi yeniden yapılandırmada iyi bir uyum sağladığını göstermektedir. Son olarak, ikinci veri setini oluşturmak için otokodlayıcı modeli kaydedilmiştir.



Şekil 3.7. Eğitilmiş Otomatik kodlayıcının train ve test sonuçları.

3.3 İkinci Veri Kümesi Oluşturma

İkinci veri setini oluşturmak için, aynı veri seti okunmuştur ve %80 train seti ve %20 test seti olarak ayırt edilmiştir. Ayrıca verileri 0 ile 1 aralığında normalleştirmek için MinMaxScaler kullanılmıştır. Ardından kayıt olan Otomatik Kodlayıcı model yüklenmiş olup, sadece kodlayıcı kısmın darboğaza kadar kullanılmıştır. Bu şekilde, 30 sütunu 16 sütuna düşürüp, orijinal veri setin üzerinde öznitelik çıkarma kullanarak, bir yandan gereksiz özellikleri atarken, modelin hızı da artırmak amaçlanmıştır. Yukarıdaki açıklamanın python ile geliştirme kodu şekil 3.8'de gösterilmiştir.

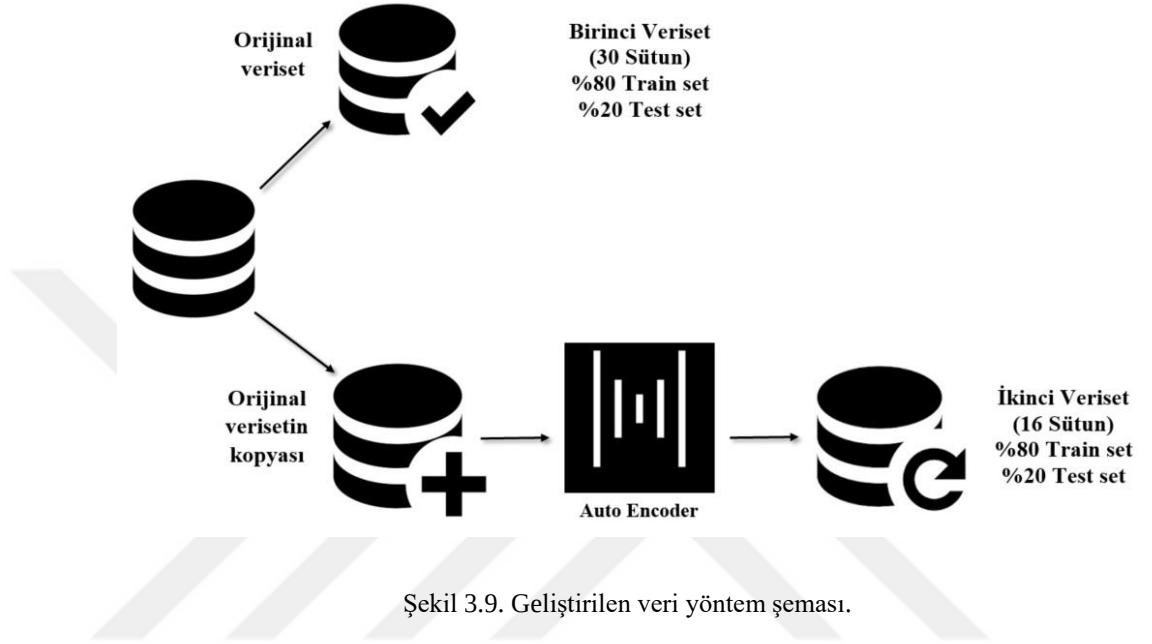
```

105 encoder.save(dir_path + '\\RBM\\encode_ds\\encoder_rbm.h5')
106
107 ds_train = pd.read_csv(dir_path + '\\RBM\\train_rbm.csv')
108 ds_test = pd.read_csv(dir_path + '\\RBM\\test_rbm.csv')
109 # load the model from file
110 encoder = load_model(dir_path + '\\RBM\\encode_ds\\encoder_rbm.h5', compile=False)
111
112 # encode the train data
113 ds_train.drop(['Time'], axis=1, inplace=True)
114 #print("ds_train shape 0 is ", ds_train.shape[0], " shap 1 is ", ds_train.shape[1])
115 y_train_enc = ds_train.iloc[:, -1].values
116 ds_train = t.transform(ds_train.iloc[:, :-1].values)
117 X_train_encode = encoder.predict(ds_train)
118 #X_train_encode = ds_train.iloc[:, :-1].values
119
120
121 df_merge_train = pd.merge(pd.DataFrame(X_train_encode), pd.DataFrame(y_train_enc), how='left', left_index= True,
122 df_merge_train.to_csv(dir_path + '\\RBM\\train_rbm_encode.csv', index = False, header=True)
123
124 # encode the test data
125 ds_test.drop(['Time'], axis=1, inplace=True)
126 #print("ds_test shape 0 is ", ds_test.shape[0], " shap 1 is ", ds_test.shape[1])
127 y_test_enc = ds_test.iloc[:, -1].values
128 ds_test = t.transform(ds_test.iloc[:, :-1].values)
129 X_test_encode = encoder.predict(ds_test)
130
131 df_merge_test = pd.merge(pd.DataFrame(X_test_encode), pd.DataFrame(y_test_enc), how='left', left_index= True,
132 df_merge_test.to_csv(dir_path + '\\RBM\\test_rbm_encode.csv', index = False, header=True)
133

```

Şekil 3.8. Otomatik Kodlayıcının model yükleme ve ikinci veriset oluşturma Python kodu.

Geliştirmelerin sonucunda 1. Veri setimiz orjinal şekilde 30 sütunla, kullanmaya hazırlanırken, 2. Veri setimiz ön işlemden sonra elde etmiş olup 16 sütundan oluşmaktadır. Geliştiren veri yöntemin şeması Şekil 3.9’da sunulmuştur.



4. YÖNTEM

Kredi kartı dolandırıcılık tespiti için en iyi 2 derin öğrenme yöntemleri kullanılmıştır ve ardından bu modellerin avantajlarını bir araya getirmek adına topluluk model geliştirilmiştir. Bu tez çalışmasında temel sınıflandırma yöntemleri olarak Kısıtlı Boltzmann Makinesi, Kısa Uzun Süre Bellek ve topluluk için Soft-Voting Bagging yapılar oluşturulmuştur. Bu yöntemler aşağıdaki alt başlıklarda açıklanmaktadır.

4.1 Kısıtlı Boltzmann Makinesi

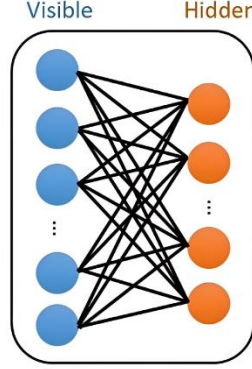
Kısıtlı Boltzmann Makinesi, 1985 yılında Geoffrey Hinton tarafından icat edilmiştir ve gözetimsiz öğrenme algoritmaların kategorisine girmektedir. Stokastik kararlar veren simetrik olarak bağlı nöron benzeri birimlerden oluşan bir ağıdır. Bu derin öğrenme algoritması, filmler için kullanıcı puanlarını tahmin etmek ve rakiplerinin çoğunu yenmek için RBM'nin ortak bir filtreleme tekniği olarak kullanıldığı Netflix Yarışmasından sonra çok popüler oldu. Regresyon, sınıflandırma, boyutluluk azaltma, özellik öğrenme, konu modelleme ve işbirlikçi filtreleme için kullanışlıdır (Medium, 2021).

4.1.1 Kısıtlı Boltzmann Makinenin özellikleri

Kısıtlı Boltzmann Makineler, girdiyi yeniden yapılandırarak verilerdeki doğal kalıpları otomatik olarak algılayabilen bir enerji tabanlı model kategorisine ait olan stokastik iki katmanlı sinir ağlardır. İki katmanı vardır: görünür ve gizli. Görünür katmanın girdi düğümleri vardır ve gizli katman, verilerden özellik bilgilerini çıkaran düğümlerden oluşur. Gizli katmandaki çıktı, girdi katmanlarının ağırlıklı toplamıdır. Modelde herhangi bir çıktı düğümü yoktur ve ayrıca kalıpların öğrenildiği tipik ikili çıktıya sahip değildir. Yalnızca giriş düğümleriyle ilgilenir. Giriş sağlandıktan sonra, RBM'ler veriler arasındaki tüm kalıpları, parametreleri ve korelasyonları otomatik olarak yakalar. Şekil 4.1'de modelin genel yapısı sunulmuştur.

- RBM, enerji tabanlı modele ait sinir ağıdır.
- Olasılığa dayalı, gözetimsiz, üretken bir derin makine öğrenme algoritmasıdır.
- RBM'nin amacı, log-olabilirlik fonksiyonunu maksimize eden ortak olasılık dağılımını bulmaktır.
- RBM yönsüzdür ve yalnızca iki katmana sahiptir: Görünür katmanı ve gizli katman.

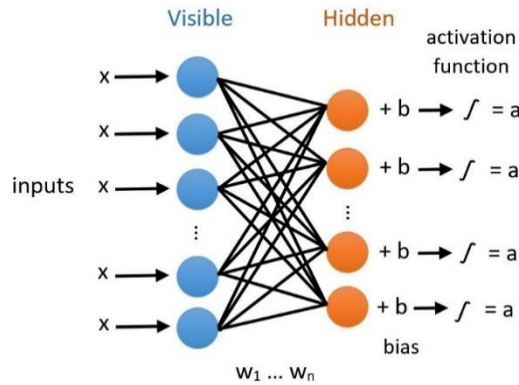
- Tüm görünür düğümler, tüm gizli düğümlere bağlanır. Ne görünür ne de gizli katmanın katman içi bağlantısı yoktur. Yalnızca girdi ve gizli düğümler arasında bağlantılar vardır, bu nedenle asimetrik iki parçalı çizge olarak da adlandırılır.



Şekil 4.1. RBM modelin genel yapısı.

4.1.2 Kısıtlı Boltzmann Makinenin çalışma şekli

RBM, bir Stokastik Sinir Ağıdır, bu, her nöronun etkinleştirildiğinde bazı rastgele davranışlara sahip olacağı anlamına gelir. Önyargı birimlerinin iki katmanı daha vardır: gizli sapma ve görünür sapma. Bu, RBM'leri Otomatik Kodlayıcılardan farklı kılan şeydir. Gizli önyargı RBM, ileri geçişte aktivasyonu üretir ve görünür önyargı, geri geçiş sırasında RBM'nin girişi yeniden yapılandırmasına yardımcı olur. Görünür birimler arasında bağlantı olmadığından ve bu nedenle kendi aralarında bilgi aktarmanın bir yolu olmadığından, yeniden oluşturulan girdi gerçek girdiden her zaman farklıdır.

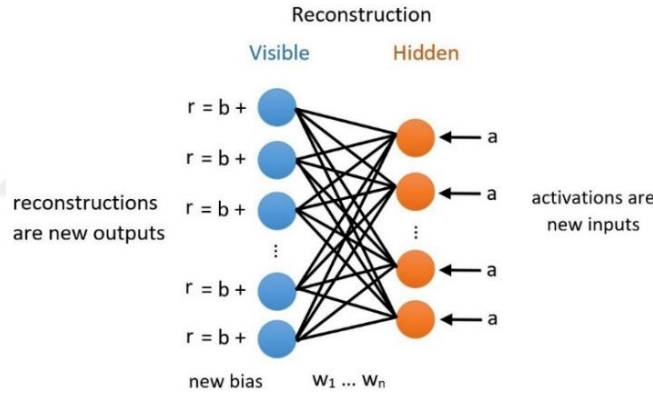


Şekil 4.2. RBM eğitimindeki ilk aşaması.

Şekil 4.2'deki görüntü, birden çok girdiye sahip bir RBM'nin eğitimindeki ilk adımı göstermektedir. Girdiler ağırlıklarla çarpılır ve ardından sapmaya eklenir. Sonuç daha sonra bir sigmoid aktivasyon fonksiyonundan geçirilir ve çıktı, gizli durumun etkinleştirilip etkinleştirilmediğini belirler. Ağırlıklar, satır sayısı olarak giriş düğümlerinin sayısı ve sütun sayısı olarak gizli düğümlerin sayısı ile bir matris olacaktır. İlk gizli düğüm, karşılık gelen önyargı terimi eklenmeden önce ilk ağırlık sütunu ile çarpılan girdilerin vektör çarpımını alacaktır. Yani bu adımda elde ettiğimiz denklem aşağıdaki gibidir:

$$h^{(1)} = S(v^{(0)T}W + a) \quad (4.1)$$

Burada $h(1)$ ve $v(0)$, gizli ve görünür katmanlar için karşılık gelen vektörlerdir veya başka bir deyişle sütun matrisidir, yineleme $v(0)$ olarak üst simge ile birlikte, ağa sağladığımız girdi anlamına gelir ve a , gizli katmanın önyargı vektörüdür.



Şekil 4.3. RBM öğrenmenin yeniden yapılandırma aşaması.

Şekil 4.3'deki görüntü ters aşamayı veya yeniden yapılandırma aşamasını gösteriyor. İlk geçişe benzer, ancak ters yöndedir. Denklem şu şekildedir:

$$v^{(1)} = S(h^{(1)T}W + b) \quad (4.2)$$

Burada $v(1)$ ve $h(1)$, yineleme olarak üst simge ile görünür ve gizli katmanlar için karşılık gelen vektörlerdir veya başka bir deyişle sütun matrisidir ve b , görünür katman önyargı vektörüdür. $v(0) - v(1)$ farkı, eğitim sürecinin sonraki adımlarında azaltmamız gereken yeniden yapılandırma hatası olarak düşünülebilir. Bu nedenle, bu hatayı en aza indirmek için ağırlıklar her yinelemede ayarlanır. İleri geçişte, $h(1)$ çıktısının olasılığını hesaplamak için $v(0)$ girdisi ve W ağırlıkları verdiğinde aşağıdaki şekilde gösterilir:

$$p(h^{(1)}|v^{(0)}; W) \quad (4.3)$$

Ve geriye doğru geçişte, girdiyi yeniden yapılandırırken, $h(1)$ girdisi ve W ağırlıkları ile verilen $v(1)$ çıktısının olasılığını hesaplıyoruz:

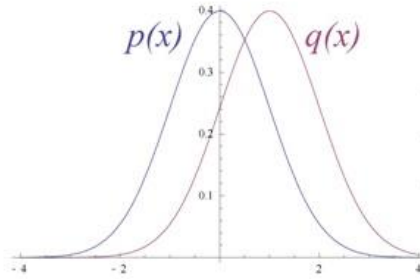
$$p(v^{(1)}|h^{(1)}; W) \quad (4.4)$$

Hem ileri hem de geri geçişte kullanılan ağırlıklar aynıdır. Birlikte, bu iki koşullu olasılık, girdilerin ve aktivasyonların ortak dağılımına götürmektedir.

4.1.3 Gibbs sampling ve Contrastive Divergence

RBM'ler yönsüz olduğundan, ağırlıklarını gradyan iniş ve geri yayılım yoluyla ayarlamazlar. Ağırlıklarını, karşılaştırmalı sapma adı verilen bir süreçle ayarlarlar. Bu sürecin başlangıcında, görünür düğümlerin ağırlıkları rastgele oluşturulur ve gizli düğümleri oluşturmak için kullanılır. Bu gizli düğümler daha sonra görünür düğümleri yeniden oluşturmak için aynı ağırlıkları kullanır. Görünür düğümleri yeniden oluşturmak için kullanılan ağırlıklar baştan sona aynıdır. Ancak oluşturulan düğümler birbirine bağlı olmadığı için aynı değildir.

Yeniden yapılandırma, bir girdi örneğine sürekli/ayrık bir değer ilişkilendirmek yerine orijinal girdinin olasılık dağılımını tahmin etmesi bakımından regresyon veya sınıflandırmadan farklıdır. Bu, aynı anda birden fazla değeri tahmin etmeye çalıştığı anlamına gelir. Bu, bir sınıflandırma probleminde (girişi etiketlere eşleme) meydana gelen ayırt edici öğrenmenin aksine, üretken öğrenme olarak bilinir.



Şekil 4.4. Öğrenme sürecini gerçek ve tahminde bulunan dağılım örneği.

Şekil 4.4'te görüldüğü gibi $p(x)$ gerçek bir modelin dağılımıyken, $q(x)$ tahmin edilen dağılımdır ve amaç bu $q(x)$ 'i $p(x)$ 'e yanaştırmaktır. Gibbs örnekleme veya Gibbs örnekleme, doğrudan örnekleme zor olduğunda belirli

bir çok değişkenli olasılık dağılımından yaklaşık olarak elde edilen bir dizi gözlem elde etmek için bir MCMC algoritmasıdır. Bu dizi, ortak dağılıma yaklaşmak için kullanılabilir (örneğin, dağılımın bir histogramını oluşturmak için); değişkenlerden birinin veya değişkenlerin bazı alt kümelerinin (örneğin, bilinmeyen parametreler veya gizli değişkenler) marjinal dağılımına yaklaşmak için; veya bir integrali hesaplamak için (değişkenlerden birinin beklenen değeri gibi). Log-olasılık gradyanının Gibbs örneklemeyle dayalı stokastik yaklaşımlarına dayanan öğrenme algoritmaları, Kısıtlı Boltzmann Makinelerini eğitmenin yaygın bir yolu haline gelmiştir.

Kontrastlı sapma ise parametreleri güncellemek için gereken RBM'lerdeki gradyanın inatçı negatif fazının tahminidir. Negatif fazın kesin hesaplanması zor olduğundan, negatif fazdaki toplamı tahmin etmek için MCMC entegrasyonuna başvurulur. Aslında, Hinton ve ark. Durağanlığı beklemeden sadece birkaç sabit sayıda MCMC yinelemesinin, negatif fazın adil bir tahminini almak ve bir gradyan güncellemesi yapmak için yeterli olduğunu buldu. RBM'lerde gradyan güncellemesi yapmanın bu yolunu kontrastlı sapma olarak adlandırdılar.

RBM- Contrastive Divergence Algoritması

Girdi: Her bir işlem için sınıf değeri ve öznelik değerlerini içeren vektör. (X^t)

Çıktı: Öznelikler ile ilgili tahminler içeren vektörü. ($X^{\bar{t}}$)

1. **For** each training example X^t
 2. Generate a negative sample $\bar{X}$
 3. k steps of Gibbs sampling, starting at X^t
 4. Update parameters
 5. $W \leftarrow W + \alpha (h(x^t x^{t^T}) - h(\bar{x}) \bar{x}^T)$
 6. $b \leftarrow b + \alpha (h(x^t) - h(\bar{x}))$
 7. $c \leftarrow c + \alpha (x^t - \bar{x})$
 8. **Go to 1 until stopping criteria**
-

4.1.4 Kredi kartı dolandırıcılık için RBM Modeli geliştirme

Python'da RBM modeli implemente etmek için TensorFlow kütüphanesi kullanılmıştır. Bu model'de aşağıdaki girdileri içeren RBM adında bir class oluşturulmuştur:

num_visible: Bu değişken, modelimizde görünen katman sayısıdır. İki farklı veri setimiz olduğundan dolayı, birinci veriset için (orijinal olan) değeri 30'a eşittir, çünkü 30 sütun vardır ve ikinci veri set ise 15 sütun varlığından dolayı 15'e set edilmiştir.

num_hidden: RMB modelindeki gizli katmanların sayısını belirlemektedir. Birinci veri seti için 15, ikincisi için 10 olarak belirlenmiştir.

visible_unit_type: Adından da anlaşılacağı gibi, Görünür birim tipini tanımlar, ayrıca veri setindeki dolandırıcılık ve normal işlemlerin sürekli değerleri ile uğraştığımızdan dolayı, gauss tipi olarak ayarlanmıştır.

main_dir: Modelin çalıştığı ve kayıt olacak yönünü belirlemektedir.

model_name: Geliştirdiğimiz modelin ismini belirlemektedir. 'rbm_model' adına kayıt edilmiştir.

gibbs_sampling_step: Daha önce açıklandığı gibi, Gibbs örnekleme veya bir Gibbs örnekleme, doğrudan örnekleme zor olduğu durumlarda, belirli bir çok değişkenli olasılık dağılımından yaklaşık olarak elde edilen bir dizi gözlemi elde etmek için bir MCMC algoritmasıdır. Her Gibbs örnekleme adımı, tüm gizli ve tüm görünür birimleri sırayla güncellemek anlamına gelir. Log-olasılık gradyanının Gibbs örnekleme tabanlı stokastik yaklaşımlarına dayanan öğrenme algoritmaları, Kısıtlı Boltzmann Makinelerini eğitmenin yaygın bir yolu haline geldi. Bu, öğrenme süreci boyunca ağırlıkları ayarlayan Kontrastlı Iraksamadır.

learning_rate: öğrenme oranı, minimum kayıp fonksiyonuna doğru hareket ederken her yinelemede adım boyutunu belirleyen bir optimizasyon algoritmasındaki bir ayar parametresidir.

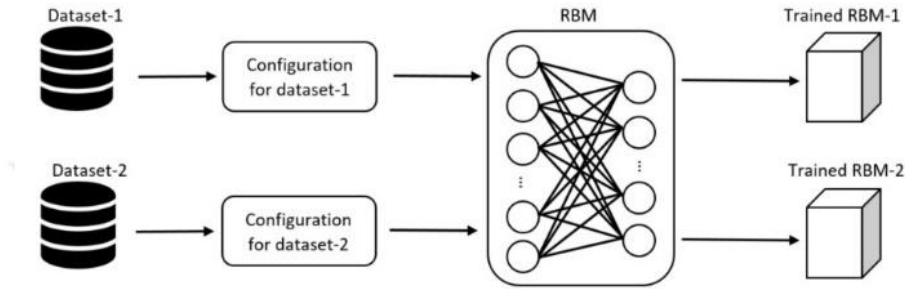
momentum: Geri yayılım sırasında, en son eğitim sonuçlarına uyum sağlamak için modelin ağırlıklarını ayarlamaktadır. Momentum, tutarlı bir yönü korumak için benzer bir girişimdir. Daha küçük adımlar atılıyorsa, alanda biraz tutarlı bir ilerleme sağlamak da mantıklıdır.

batch_size: parti boyutu, bir defada ağa geçirilecek numune sayısıdır.

num_epochs: öğrenme sırasındaki dönem (yineleme) sayısını tanımlamaktadır.

Şekil 4.5'te gösterildiği gibi, RBM modelini veri seti-1 ve veri seti-2'nin eğitim seti (%80) ile ayrı ayrı eğitilmiş olup daha sonra nihai karar destek sistemi için kullanmak üzere kaydedilmiştir. Aslında RBM, işlem başına 0 ve 1 nihai

sonucunu tahmin etmek yerine, tüm gerçek işlemlerden oluşan tren seti verilerinden normal bir dağılım yapmaya çalışmaktadır. Tüm hileli ve normal işlemlerden oluşan test setinde model, normal dağılımın dışında yer alan anormalliği bulmaya çalışmaktadır ve eğer bulunursa bunun hileli bir işlem olarak kabul edilebileceğini söyleyebiliriz. Bahse geçen şekilde konfigürasyonların ayrı şekilde çizilmesi, birinci ve ikinci veri setinde birbirlerine çok benzer ancak denemelerde ufak bir farkla en iyi sonuç alan şekilde kayıt edildiğinden kaynaklıdır.



Şekil 4.5. Her iki veri kümesi için RBM sınıflandırıcısının şeması.

```

model = RBM(train_x.shape[1], 24, visible_unit_type='gauss', main_dir=dir_path, model_name='rbm_model.ckpt',
            gibbs_sampling_steps=4, learning_rate=0.005, momentum = 0.95, batch_size=64, num_epochs=10, verbose=1)

# Perform 10-fold cross-validation
kfold = KFold(n_splits=10, shuffle=True, random_state=42)
accuracies = []

for train_index, val_index in kfold.split(train_x):
    X_train_fold, X_val_fold = train_x[train_index], train_x[val_index]
    y_train_fold, y_val_fold = train_y[train_index], train_y[val_index]

    # Train the model on the training fold
    model.fit(X_train_fold, validation_set=X_val_fold, #y_train_fold)

    # Make predictions on the validation fold
    y_pred = model.getFreeEnergy(X_val_fold).reshape(-1)#.predict(X_val_fold)

# Print the average accuracy across all folds
#print("Average accuracy:", sum(accuracies) / len(accuracies))

model.fit(train_x, validation_set=test_x)
test_cost = model.getFreeEnergy(test_x).reshape(-1)

test_cost = test_cost.round(2)

# Check for NaN values
nan_values = np.isnan(test_cost).any() or any(x is None for x in test_cost)
if nan_values:
    print("There is NaN values")
else:
    print("Every value is OK")
  
```

Şekil 4.6. Python ile geliştirilmiş RBM modeli.

Python uygulamasında bahsedilen tüm parametreler şekil 4.6'da gösterilmektedir. Tren setin üzerinde modele uygun bir girdi vermek adına

verilerde ön işlemler yapılmıştır. Ardından, RBM modelini çağırıp parametreleri ayarladıktan sonra, aşırı uyumu önlemek ve sağlam performans tahmini için, tren setleri 10-katlı çapraz doğrulama kullanarak model eğitilip “rbm_model.ckpt” olarak kaydedilmiştir. Son aşamada, test setin üzerinde bir tahmin yapılmıştır.

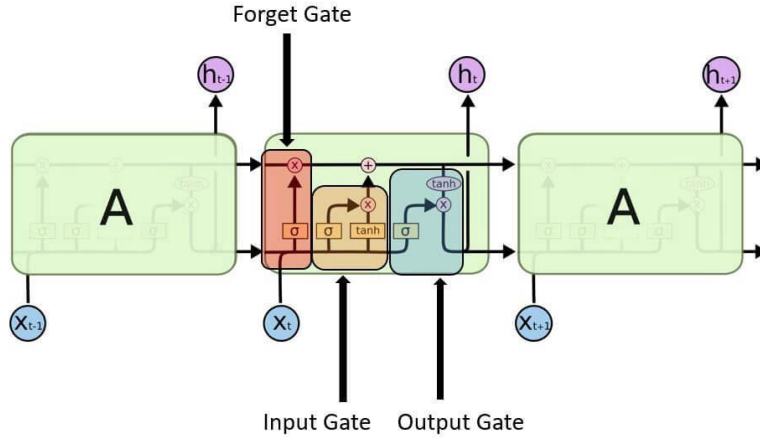
4.2 Kısa Uzun Süre Bellek

Uzun Kısa Süreli Bellek, Hochreiter & Schmidhuber tarafından tasarlanan bir tür tekrarlayan sinir ağıdır. RNN'de son adımın çıkışı, mevcut adımda giriş olarak beslenir. RNN'nin uzun süreli bellekte saklanan kelimeyi tahmin edemediği ancak son bilgilerden daha doğru tahminler verebildiği RNN'nin uzun vadeli bağımlılıkları sorununu ele almaktadır. Boşluk uzunluğu arttıkça RNN verimli bir performans vermemektedir. LSTM, varsayılan olarak bilgileri uzun bir süre saklayabilir. Zaman serisi verileri temelinde işlemek, tahmin etmek ve sınıflandırmak için kullanılmaktadır.

Derin Tekrarlayan Sinir Ağında vanishing ve exploding Gradyanlar problemini çözmek için birçok varyasyon geliştirilmiştir. Bunların en ünlülerinden biri Uzun Kısa Süreli Bellek Ağıdır. Konsept olarak, bir LSTM tekrarlayan birimi, ağıın şimdiye kadar görüldüğü tüm geçmiş bilgileri "hatırlamaya" ve alakasız verileri "unutmaya" çalışır. Bu, farklı amaçlar için “kapılar” adı verilen farklı aktivasyon fonksiyon katmanlarının tanıtılmasıyla yapılır. Her LSTM tekrarlayan birimi aynı zamanda, önceki LSTM tekrarlayan birimi tarafından muhafaza edilmek üzere seçilen bilgiyi kavramsal olarak tanımlayan Dahili Hücre Durumu adı verilen bir vektörde muhafaza eder (Machine learning mastery, 2020).

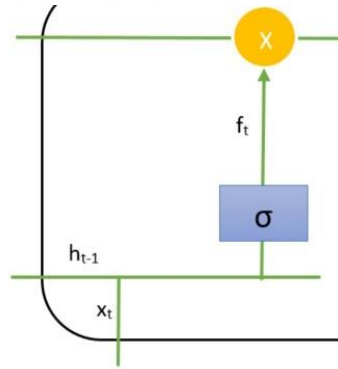
4.2.1 LSTM çalışma şekli

LSTM, dört sinir ağından ve bir zincir yapısında hücreler olarak bilinen çok sayıda bellek bloğundan oluşur. Geleneksel bir LSTM birimi, bir hücre, bir giriş kapısı, bir çıkış kapısı ve bir unutma kapısından oluşur. Hücrenin içine ve dışına bilgi akışı üç kapı tarafından kontrol edilir ve hücre keyfi zaman aralıklarında değerleri hatırlar. LSTM algoritması, belirsiz süreli zaman serilerini kategorize etmek, analiz etmek ve tahmin etmek için iyi bir şekilde uyarlanmıştır. Şekil 4.7’de gördüğünüz gibi LSTM’in üç girişi vardır (Geeks for geeks, 2023):



Şekil 4.7. LSTM yapısının detayı.

Unut kapısı (forget gate): Bir unut kapısı, hücre durumundan bilgileri kaldırmaktan sorumludur. LSTM'in anlaması için artık gerek olmayan bilgileri veya daha az önemli olan bilgiler, bir filtrenin çarpılması yoluyla kaldırılmaktadır. Bu, LSTM ağıнын performansını optimize etmek için gereklidir. Şekil 4.8'de görüldüğü gibi, bu kapı iki giriş alır: h_{t-1} ve x_t .

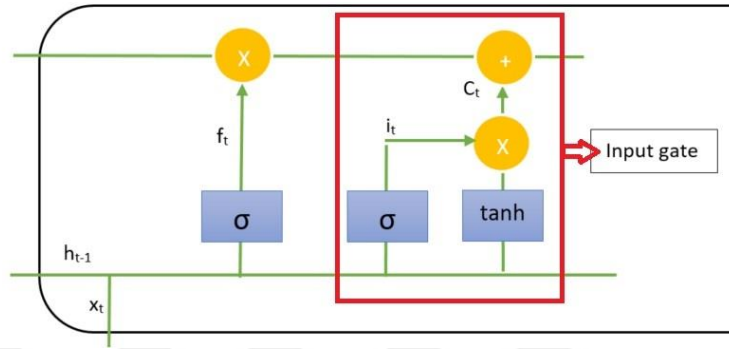


Şekil 4.8. LSTM'in unut kapısının detayı.

h_{t-1} , önceki hücrenin gizli durumu veya önceki hücrenin çıktısıdır ve x_t , belirli zaman adımındaki girdisidir. Verilen girdiler ağırlık matrisleri ile çarpılır ve bir sapma eklenir. Bunu takiben, bu değere sigmoid işlevi uygulanır. Sigmoid işlevi, hücre durumundaki her sayıya karşılık gelen '0' ile '1' arasında değişen değerlere sahip bir vektör çıkarır. Temel olarak, sigmoid işlevi, hangi değerlerin tutulacağına ve hangilerinin atılacağına karar vermekten sorumludur. Hücre durumundaki belirli bir değer için bir '0' çıktıysa, bu, unutma kapısının hücre durumunun bu bilgi parçasını tamamen unutmasını istediği anlamına gelir.

Benzer şekilde, '1', unutmaya kapısının tüm bilgiyi hatırlamak istediği anlamına gelir. Sigmoid fonksiyonundan gelen bu vektör çıktısı, hücre durumu ile çarpılır.

Giriş kapısı (input gate): hücre durumuna bilgi eklenmesinden sorumludur. Bu bilgi eklenmesi, Şekil 4.9'da görüldüğü gibi temel olarak üç aşamalı bir süreçtir.

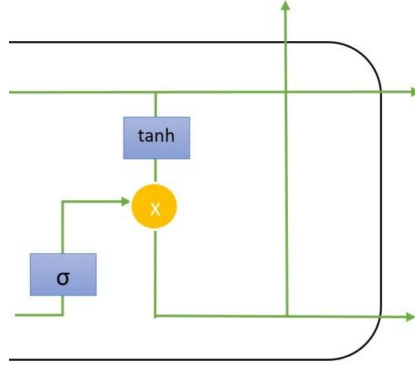


Şekil 4.9. LSTM'ın giriş kapısının detayı.

1. Bir sigmoid işlevi dahil ederek hücre durumuna hangi değerlerin eklenmesi gerektiğini düzenlemektedir. Bu, unutmaya kapısına çok benzer ve h_{t-1} ve x_t 'den gelen tüm bilgiler için bir filtre görevi görür.
2. Hücre durumuna eklenebilecek (h_{t-1} ve x_t 'den algılandığı gibi) tüm olası değerleri içeren bir vektör oluşturmaktadır. Bu, '-1' ile '+1' arasındaki değerleri veren tanh işlevi kullanılarak yapmaktadır.
3. Düzenleyici filtrenin veya başka bir deyişle sigmoid geçidi, değerinin oluşturulan vektörle yani tanh fonksiyonu çarparak, ardından bu faydalı bilginin toplama işlemi ile hücre durumuna eklemektedir.

Bu üç aşamalı işlem tamamlandıktan sonra, hücre durumuna yalnızca önemli olan ve gereksiz olmayan bilgilerin eklenmesini sağlıyoruz.

Çıkış kapısı (output gate): Bir çıkış kapısının işleyişi, Şekil 4.10'da görüldüğü gibi, yine üç adıma ayrılabilir:



Şekil 4.10. LSTM’ın çıkış kapısının detayı.

1. Hücre durumuna tanh işlevi uyguladıktan sonra bir vektör oluşturup, böylece değerleri ‘-1’ ile ‘+1’ aralığına scale edebilmektedir.
2. h_{t-1} ve X_t değerlerini kullanarak, yukarıda oluşturulan vektörden çıktı alınması gereken değerleri düzenleyebilecek şekilde bir filtre yapmaktadır. Bu filtre yine bir sigmoid işlevini kullanır.
3. Bu düzenleyici filtrenin değerinin birinci adımda oluşturulan vektörle çarpıp onu hem çıktı olarak, hem bir sonraki hücrenin gizli durumuna göndermektedir.

4.2.2 Kredi kartı dolandırıcılık için LSTM model geliştirilmesi

LSTM modelini geliştirmek için python programlama dilinin Keras kütüphanesi kullanılmıştır. Bu nedenle, LSTM modelini oluşturmak için şekil 4.11’de görüldüğü gibi Keras’tan birkaç modül almamız gerekiyor.

- Sinir ağını başlatmak için “Sequential”
- Yoğun ve bağlı bir sinir ağı eklemek için “Dense”
- Uzun Kısa Süreli Bellek katmanını eklemek için “LSTM”
- overfittin’i önlemek için “Dropout”

```
22 from keras.models import Sequential
23 from keras.layers import Dense, LSTM, Dropout
```

Şekil 4.11. LSTM için keras kütüphanesi kullanımı.

Daha önce de belirttiğimiz gibi, kredi kartı dolandırıcılık tespiti için iki LSTM katmanı ve bir Yoğun (Dense) katman ile sıralı LSTM modeli uygulanmıştır. Geliştirme python kodu şekil 4.12’de sunulmuştur.

```
## Reshape input to be 3D [samples, timesteps, features] (format requires per LSTM)
X_train = train_x.reshape((train_x.shape[0], 1, train_x.shape[1]))
X_test = test_x.reshape((test_x.shape[0], 1, test_x.shape[1]))

### Train LSTM using Keras 2 API ###
model = Sequential()
model.add(LSTM(128, input_shape=X_train.shape[1:], kernel_initializer='lecun_uniform', activation='relu', kernel_regularizer=
#model.add(LSTM(12, activation='relu', return_sequences=True))
#model.add(LSTM(100, activation='relu', return_sequences=True))
model.add(LSTM(128, activation='relu'))
model.add(Dense(1, kernel_initializer='lecun_uniform', activation='sigmoid'))

history = model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy']) #optimizer='rmsprop'
print(model.summary())

# Perform 10-fold cross-validation
kfold = KFold(n_splits=10, shuffle=True, random_state=42)
accuracies = []

for train_index, val_index in kfold.split(X_train):
    X_train_fold, X_val_fold = X_train[train_index], X_train[val_index]
    y_train_fold, y_val_fold = y_train[train_index], y_train[val_index]

    # Train the model on the training fold
    model.fit(X_train_fold, y_train_fold, epochs=10, batch_size=256)

    # Make predictions on the validation fold
    y_pred = model.predict_classes(X_val_fold)

# Print the average accuracy across all folds
print("Average accuracy:", sum(accuracies) / len(accuracies))

model.fit(X_train, y_train, epochs=30, batch_size=256, validation_split=0.3)

#train_y = train_y.reshape((train_y.shape[0], 1, 30))
print(X_train.shape)
#split_input(train_y)
print(y_train.shape)
```

Şekil 4.12. LSTM modelini python kodu ile geliştirilmesi.

Yukarıdaki şekilde gördüğümüz gibi, LSTM model’i giriş şekli nedeniyle, tren ve test veri setini [sample, time steps, features] 3-Boyutlu formatında yeniden şekillendirilmiştir:

Sample: veri kümesinin satır sayısı olacaktır.

Time steps: bir diziyi beslediğiniz zaman adımlarının sayısını temsil eder. Zincirin kaç adımını dikkate almamız gerektiğini söylüyor. Örneğin modelimizde ‘1’ olarak ayarlanmıştır. Bu, LSTM bloğu öğrenme sürecini hesaplamak istediğinde, bir önceki Blok’u dikkate aldığı anlamına gelir. ‘5’ olarak ayarlansaydı, son beş blok olarak kabul edilirdi.

Features: veri kümesindeki sütunların sayısına eşittir.

LSTM katmanını ekledikten sonra aşağıdaki argümanlarla overlap önlemek için bir Dropout katmanı eklenmiştir, ayrıca aşırı uyumu önlemek ve

sağlam performans tahmini için, tren setleri 10-katlı çapraz doğrulama kullanılmıştır:

Çıkış uzayının boyutu olan 128 birimli LSTM kullanılmıştır.

`return_sequences=True`, çıktı dizisindeki son çıktının mı yoksa tam dizilimdeki son çıktının mı döndürüleceğini belirler.

Dropout katmanlarını tanımlarken, katmanların %20'sinin bırakılacağı anlamına gelen '0.2' değeri belirlenmiştir. Modelimiz için `relu` aktivasyon fonksiyonu kullanılmıştır.

'1' birimin çıktısını belirten Dropout katman eklenmiştir. İkili problemlerle uğraştığımız için, nihai sonucu tahmin etmek için sigmoid aktivasyon fonksiyonu kullanılmıştır.

sonra, popüler adam optimizier kullanarak modelimiz derlenmiş olup Loss metriğimiz "`binary_crossentropy`" ile MSE olarak ayarlanmıştır. Bu karesi alınmış hataların ortalamasını hesaplayacaktır.

LSTM modeli geliştirirken, modelin işlem zamanı bazlı olduğundan dolayı, veri setlerimiz'de işlem zamanı sırasına dikkat edilmiştir halbuki RBM için dikkat edilmemiştir. LSTM'ın pseudo kodu aşağıda getirilmiş ve modeli 30 epoch ile çalışacak şekilde fit edilmiştir. Ancak farklı veri kümeleri için en iyi sonuçları elde etmek adına, diğer parametreleri farklı ayarlarda değiştirilmiştir.

LSTM Algoritması

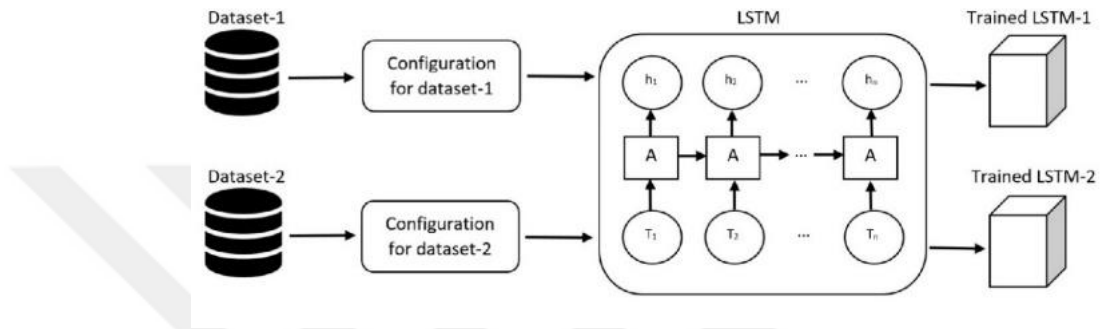
Girdi: Her bir eğitim örneği için sınıf değeri ve öznitelik değerlerini içeren x vektör.

Çıktı: işlemlerin dolandırıcılık ile ilgili tahminler içeren h vektörü.

1. Input: $x = [x_1, \dots, x_n]$;
 2. Given parameters: W, U, b for all of transaction
 3. Initialize all parameters
 4. **For** $t=1$ **to** 100 **do begin**
 5. Calculate $f_t, \bar{c}_t, i_t$
 6. Update cell state c_t
 7. Calculate o_t and h_t
 8. **End for**
-

Modelimizi hızlandırmak adına minimum sayıda katman kullanmaya çalışıp, sıralı LSTM modelini iki LSTM katmanlı ve bir yoğun katmanla tasarlanıp uygulanmıştır. LSTM modelinin giriş şekli nedeniyle, eğitim ve test veri kümesini 3-Boyutlu formatta (örnek, işlem zamanı adımları, özellikler)

yeniden şekillendirilmiştir. “Örnek” veri kümesinin satır sayısına aittir. "İşlem adımları", bir diziyi beslediğiniz gerçekleşme adım sayısını temsil eder. Zincirin kaç adımını dikkate almamız gerektiğini söylemektedir. “Özellikler” veri kümesindeki sütun sayısına eşittir. Şekil 4.13'te LSTM sınıflandırıcının şemasını görüldüğü gibi, modeli her iki veri setine dayalı olarak eğitilmiştir ve bunları son topluluk modelinin temel sınıflandırıcıları için kaydedilmiştir.



Şekil 4.13. Her iki veri kümesi için LSTM sınıflandırıcısının şeması.

4.3 Topluluk Sınıflandırıcılar

Topluluk Sınıflandırıcı olarak adlandırılan çoklu sınıflandırma sistemleri, işlemsel zekâ ve makine öğrenmesinde artan bir ilgi görmüştür. Topluluk Sınıflandırıcılar, çeşitli problemlerde ve gerçek dünya uygulamalarında çok etkili ve son derece çok yönlü olduklarını kanıtladığından bu ilgiyi hak etmiştir. Topluluk Sınıflandırıcılar, daha iyi sonuçlar elde etmek için birden çok makine öğrenimi modelinin birleştirildiği süreçtir. Bir Topluluk Sınıflandırıcı'nın kullanmanın ana nedeni, tekli modellerin hepsini bir araya getirip tek güçlü model oluşturmaktır. Topluluk Sınıflandırıcıların kullanmanın önemli faydalarından (ProjectPro, 2023):

- **Kesinlik iyileştirmek:** Topluluk sınıflandırıcı algoritmaları genellikle tek modellerden daha yüksek kesinlik elde edebilmektedirler.
- **Overfitting Azaltma:** topluluk Sınıflandırıcı algoritmaları, tekli modellerde ortaya çıkabilen bir problem olan overfitting azaltmaya yardımcı olabilir.

- **Robustness arttırmak:** Topluluk Sınıflandırıcı algoritmaları, tek modellere göre gürültüye ve verilerdeki aykırı değerlere karşı daha dayanıklıdır.
- **Interpretability geliştirmek:** Topluluk Sınıflandırıcı algoritmaları, tekli modellerden daha fazla yorumlanabilir olabilir; bu, bir makine öğrenimi modelinin sonuçlarının debugging ve anlaşılması için yardımcı olabilir.

Topluluk Sınıflandırıcı faydalarından yararlanırken bazı zorluklar ile de karşı karşıyız:

- **Complexity artışı:** Topluluk Sınıflandırıcı algoritmalarının uygulanması ve eğitilmesi, tek modellere göre daha karmaşık olabilir.
- **Hesaplama maliyetin artışı:** Topluluk Sınıflandırıcı algoritmalarının eğitilmesi, tek modellere göre hesaplama açısından daha pahalı olabilir.
- **Varyans artışı:** Topluluk Sınıflandırıcı algoritmaları, tek modellere göre daha değişken olabilir, bu da tahminin doğruluğunun bir modelden diğerine daha fazla değişebileceği anlamına gelir.

Bu tez çalışmasında Bagging, Boosting ve Stacking dahil olmak üzere üç farklı topluluk sınıflandırıcı karşılaştırılmıştır. Bu yöntemler aşağıdaki alt başlıklarda açıklanmaktadır.

4.3.1 Bagging

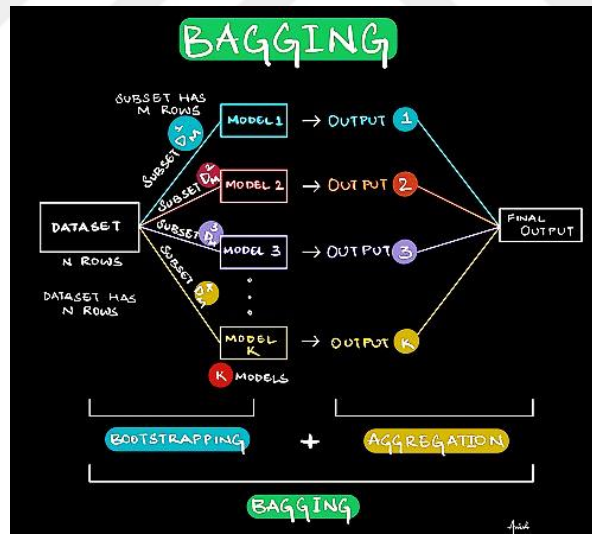
Tarihçesi 1994 yılında Leo Breiman'ın “Bagging Predictors” olarak bilinen bu algoritmayı önermiştir. Bagging’de, ilk adımda bootstrapped örnekler oluşturmaktadır. Daha sonra, her örneğe bir regresyon veya sınıflandırma algoritması uygulamaktadır. Son olarak, regresyon durumunda, bireysel öğrenenler tarafından tahmin edilen tüm çıktılar üzerinden bir ortalama alınmaktadır. Sınıflandırma için ya en çok oy alan sınıf kabul etmektedir ki hard-

voting olarak tanımlanır, ya da tüm sınıf olasılıklarının en yüksek ortalaması çıktı olarak alınmaktadır ki bunu soft-voting olarak adlandırır ve burada toplama devreye girer. Matematiksel olarak, Bagging aşağıdaki formülle temsil etmektedir:

$$\hat{f}_{bag} = \hat{f}_1(x) + \hat{f}_2(x) + \dots + \hat{f}_n(x) \quad (4.5)$$

Yukarıdaki formül’de sol taraftaki terim bagging tahmindir ve sağ taraftaki terimlerin bireysel öğrenicileridir.

Bagging veya bootstrap toplama, eğitim verilerinin farklı alt kümelerinde birden fazla modelin eğitilmesini ve bunların tahminlerinin birleştirilmesini içermektedir. Bagging, aşırı hesaplama kaynaklarını kontrol altında tutmak için paralel olarak yapılabilmektedir. Bu, beraberinde gelen iyi bir avantajdır ve genellikle algoritmanın çeşitli alanlarda kullanımını artırmak için bir güçlendiricidir. Modelin genel şeması Şekil 4.14’te sunulmuştur.



Şekil 4.14. Bagging sınıflandırıcı topluluğunun genel şeması (ProjetcPro, 2023).

Bagging Modelinin Avantajları:

- ✓ Bagging, önyargıyı artırmadan varyansı önemli ölçüde azaltır.
- ✓ Örnekleme bootstrapping ile yapıldığından, eğitim verilerindeki çeşitlilik nedeniyle bagging yöntemleri çok iyi çalışır.

- ✓ Ayrıca, eğitim seti çok büyükse, modeli nispeten daha küçük bir veri seti üzerinde eğiterek hesaplama süresinden tasarruf sağlayabilir ve yine de modelin doğruluğunu artırabilir.
- ✓ Küçük veri kümeleriyle de iyi çalışır.

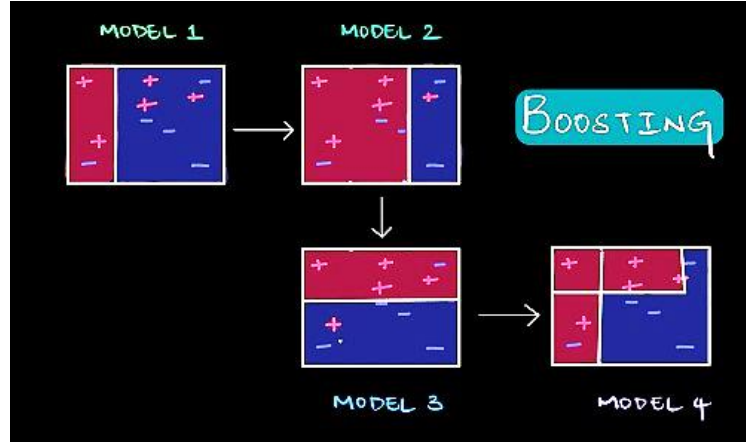
Bagging Modelinin Dezavantajları:

- Bagging’in ana dezavantajı, yorumlanabilirlik pahasına modelin doğruluğunu artırmasıdır, yani, eğer temel model olarak tek bir ağaç veya model kullanılıyorsa, o zaman daha çekici ve kolayca yorumlanabilir bir diyagrama sahip olacaktır, ancak bagging kullanıldığı zaman, bu yorumlanabilirlik kaybolur.
- Bootstrap topluluğun diğer bir dezavantajı, örnekleme sırasında hangi özelliklerin seçildiğini yorumlayamamaktadır, yani bazı özelliklerin hiç kullanılmaması ihtimali vardır, bu da önemli bilgilerin kaybolmasına neden olabilir.

4.3.2 Boosting

Boosting terimi, zayıf öğrencileri güçlü öğrencilere dönüştüren bir algoritma ailesini ifade etmektedir. Boosting, herhangi bir öğrenme algoritmasının model tahminlerini iyileştirmek için bir topluluk yöntemidir. Güçlendirme fikri, her biri bir öncekini düzeltmeye çalışan zayıf öğrencileri sırayla eğitmektedir. Zayıf öğrenenler, öncekiler tarafından sırayla düzeltilir ve bu süreçte güçlü öğrencilere dönüştürülürler.

Aslında, zayıf sınıflandırıcıların sayısından güçlü bir sınıflandırıcı oluşturmaya çalışan bir topluluk modelleme tekniğidir. Zayıf modelleri seri halinde kullanarak bir model oluşturarak yapmaktadır. İlk olarak, eğitim verilerinden bir model oluşturulur. Daha sonra birinci modelde var olan hataları düzeltmeye çalışan ikinci model kurulur. Bu prosedür, eğitim veri setinin tamamı doğru bir şekilde tahmin edilene veya maksimum model sayısı eklenene kadar devam ettirilir ve modeller eklenir. Kısaca, boosting, aynı veri kümesi üzerinde modelleri yinelemeli olarak eğitmeyi içermektedir ve sonraki her model, şekil 4.15’te görüldüğü gibi önceki model tarafından yanlış sınıflandırılan örneklerle odaklanmaktadır.



Şekil 4.15. Boosting sınıflandırıcı topluluğunun genel şeması (ProjectPro, 2023).

Boost Modelinin Avantajları:

- ✓ Boosting, aşırı uyumu kolayca engelleyen esnek bir yöntemdir.
- ✓ Kanıtlanabilir etkilidir.
- ✓ Çok yönlüdür — çok çeşitli sorunlara uygulanabilir.

Boosting Modelinin Dezavantajları:

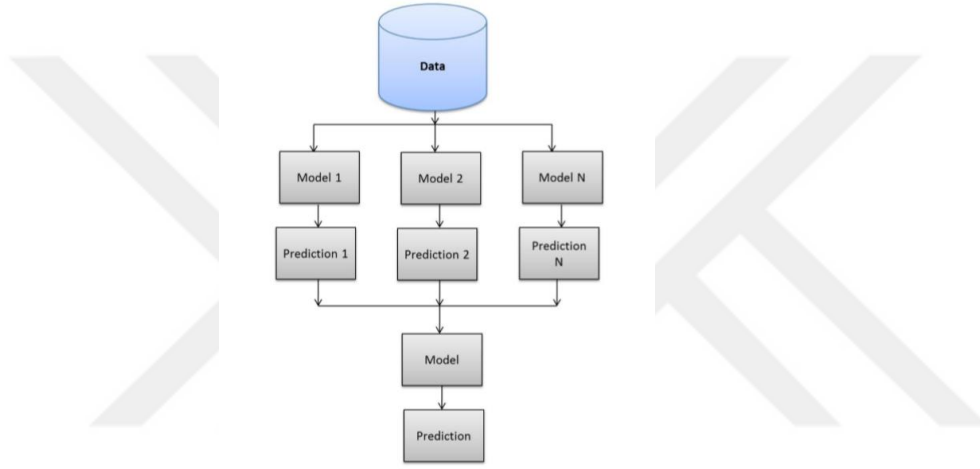
- Boosting’ın bir dezavantajı, her sınıflandırıcı öncüllerdeki hataları düzeltmek zorunda olduğundan aykırı değerlere karşı hassas olmasıdır. Bu nedenle, yöntem aykırı değerlere çok bağımlıdır.
- Diğer bir dezavantaj, yöntemin ölçeklendirilmesinin neredeyse imkansız olmasıdır. Bunun nedeni, her tahmincinin doğruluğunu önceki tahmin edicilere dayandırması ve dolayısıyla prosedürü kolaylaştırmayı zorlaştırmasıdır.

4.3.3 Stacking

Stacking, modellerini farklı öğrenme algoritmaları kullanarak oluşturan ve daha sonra temel algoritmalar tarafından üretilen tahminleri kullanarak nihai tahminler yapmak için bir birleştirici algoritma ile eğitilen bir yöntemdir. Stacking, çoklu makine öğrenimi algoritmalarını meta-öğrenme yoluyla birleştiren bir toplu öğrenme yöntemidir ki temel düzey algoritmaların eksiksiz bir eğitim veri kümesine dayalı olarak eğitildiği, meta-modelin tüm temel düzey modelin nihai sonuçları üzerinde, özellik olarak eğitilmektedir.

Stacking, temelde iki noktada bagging ve boosting'den farklıdır. Birincisi, stacking genellikle heterojen zayıf öğrencileri dikkate alırken ve farklı öğrenme algoritmaları birleştirir, bagging ve boosting ise esasen homojen zayıf öğrencileri dikkate alırlar. İkincisi, stacking, bir meta-model kullanarak temel modelleri birleştirmeyi öğrenirken, bagging ve boosting, deterministik algoritmaları izleyerek zayıf öğrencileri birleştirmektedirler.

Kısaca stacking, birden çok modelin eğitilmesini ve bunların tahminlerinin, nihai tahmini yapan nihai bir model için girdi özellikleri olarak kullanılmasını içermektedir. (şekil 4.16)



Şekil 4.16. Stacking sınıflandırıcı topluluğunun genel şeması.

Stacking topluluk model mimarisi, iki veya daha fazla temel model içeren, seviye 0 modelleri olarak etiketlenen ve seviye 1 modeli olarak bilinen bir meta-modelle yapılandırılmaktadır. Bu teknik, test ve eğitim verisi olarak işlev gören n-fold'lara ayrılmış orijinal veriyi içermektedir. Seviye 0 modelleri veya temel modeller, eğitim verisini kullanarak derlenmiş tahminler üretmektedir. Bu seviye-0 tahminleri, her bir temel modelin belirli eğitim verisi üzerinde tetiklenmesi ile oluşturulmaktadır. Stacking model mimarisi, temel modellerin tahminlerini etkili bir şekilde birleştirmek üzere tasarlanmış tek bir meta-model (seviye-1 modeli) içermektedir. Meta-model, başlangıçta eğitimleri için kullanılan veriler dışındaki verileri kullanarak bireysel temel modellerin seviye-0 tahminleri üzerinde eğitilmektedir. Bu, veriyi meta-modelin içine yerleştirme, tahminler üretme ve bu tahminleri beklenen çıktılarla birlikte kullanma sürecini içerir, bu sayede meta-modeli geliştirilmiş öngörü performansı için eğitmek için giriş ve çıkış verilerini kullanmaktadır. (JavaTPoint, 2023).

4.3.4 Topluluk Sınıflandırıcı geliřtirmesi

Yukarıdaki topluluk sınıflandırıcı modellerin açıklamalarına göre kabaca, bagging temel olarak bileřenlerinden daha az varyansa sahip bir topluluk modeli elde etmeye odaklanacağını söyleyebiliriz, oysa boosting ve stacking esas olarak bileřenlerinden daha az önyargılı güçlü modeller üretmeye (varyans da azaltılabilse bile) çalışacağını söyleyebiliriz.

Topluluk boosting modelin uygulaması karmařık ve zahmetli olabilmektedir, çünkü hiper parametrelerin dikkatli bir şekilde değerlendirilmesi, temel model seçimi, dengesiz verilerin işlenmesi ve fazla uydurmanın önlenmesi gerektirmektedir. Boosting algoritmaları, yanlış sınıflandırılmış örneklerin performansını iyileřtirmeye odaklanan yeni modellerin yinelemeli olarak eğitilmesini içermektedir. Bu yinelemeli sürecin yanında, ağırlıkların ayarlaması ve öğrenme oranlarını dikkatli ayarlama gerektirmesi, hesaplama açısından pahalıya mal olabilmektedir. Bunların yanı sıra, model'in sıralı davranıřta çalışması, bir online işlem için hileli ya da gerçek bir etiket ile karar vermesi gerektiğinde, zaman sınırimızdan dolayı uygun değildir. Bu yüzden bu topluluk sınıflandırıcı modelini kullanmayı göz ardı etmeye karar verilmiştir.

Stacking model'in, karmařıklık ve hesaplama gereksinimleri, overfitting riski, model karmařıklığını artırması, veri sızıntısı ve bilgi çakıřması problemlerinden dolayı, listemizden çıkarılmıştır.

Sonuçta, bu arařtırmada bagging topluluk sınıflandırıcı modeli, eğitilmiş RBM'in ve LSTM'in iki güçlü özelliklerini bir araya getirecek şekilde kullanılmıştır. Yani RBM'in yenilikçi dolandırıcılık bulma yeteneğini ve LSTM'in mevcut dolandırıcılık davranıřlarının algılanması avantajları, Bagging topluluk sınıflandırıcı model olarak bir araya getirilmiştir.

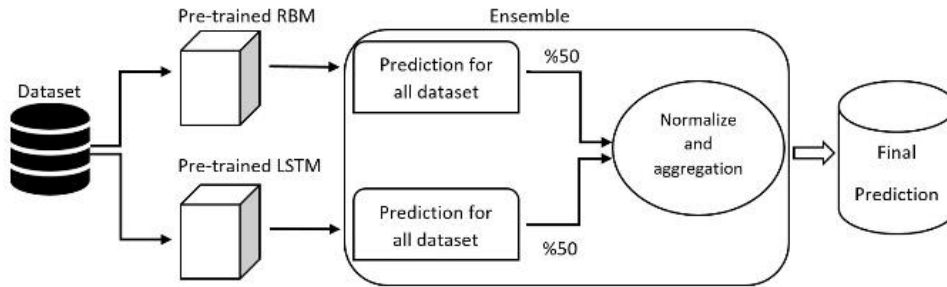
Bagging topluluk sınıflandırıcı model'i geliřtirmek için daha önce geliřtirilip kayıt edilen RBM ve LSTM modelini çağırarak, %50 şekilde, yani eşit oranda, her bir işlem için iki modelin tahmini kullanılmıştır.

Ancak iki modelin farklı yapılarından kaynaklı, LSTM modelinde tahminler 1 ile 0 şeklinde sonuçlanırken, RBM bir dağılım bazlı model olmasından dolayı geniş aralık değerleri üretmektedir. Topluluk sınıflandırma tahminin her iki modelden eşit oranda kullanabilmek adına, RBM modelinde bir

eşit belirlenmiş olup, deneysel çalışmalarda eğer elde edilen tahminler 500 üzerinde olursa, o işlem 1 yani dolandırıcılık olarak etiketlenmiştir. Böylece iki modelin tahminini %50 şekilde kullanmak adına, bir işlem için, LSTM veya RBM 1 tahmin ettiyse onu dolandırıcılık işlem olarak etiketlenmiştir. Topluluk sınıflandırıcı modelin genel şematığı Şekil 4.17’de sunulmuştur.

Kredi kartı dolandırıcılığının tespitine yönelik toplu öğrenme modelimiz kısaca aşağıdaki dört aşamadan oluşmaktadır:

- Ham verilere uygulanan bazı veri madenciliği ve makine öğrenimi tekniklerinden oluşan veri hazırlama.
- Tren veri kümelerini besleyerek DL tabanlı sınıflandırıcıların eğitimi ve güncellenmesi.
- Test veri kümesindeki temel sınıflandırıcılar için oy haklarının %50'sini dikkate alan maksimum oylama topluluğu modeli.
- Her işlem için nihai kararı vermek üzere temel sınıflandırıcının tahminlerinin normalleştirilmesi ve birleştirilmesi.



Şekil 4.17. Topluluk sınıflandırıcı modelin genel şematığı.

Avrupa kart sahipleri 2013 veri set’in tamamını, hiç değiştirmeden Birinci Final veri seti olarak, RBM ve LSTM modellerin Birinci veri set’in üzerinde eğitilmiş olanı çağırarak, bagging topluluk sınıflandırıcı modeli oluşturup, her işlem için tahminde bulunmuştur. Açıklamaların Python kodu ile geliştirmesi Şekil 4.18’da gösterilmiştir.

```

cols_mean = []
cols_std = []
for c in range(df_x_rbm.shape[1]):
    cols_mean.append(df_x_rbm[:,c].mean())
    cols_std.append(df_x_rbm[:,c].std())
    df_x_rbm[:, c] = (df_x_rbm[:, c] - cols_mean[-1]) / cols_std[-1]

model_rbm = RBM(df_x_rbm.shape[1], 15, visible_unit_type='gauss', main_dir=db_path+'RBM\\encode\\', model_name='rbm_model.ckpt',
                gibbs_sampling_steps=4, learning_rate=0.005, momentum = 0.95, batch_size=64, num_epochs=20, verbose=1)

shape = [1,2]
shape[0] = 15#29 #df.shape[0]
shape[1] = 15

model_rbm.load_model(shape, 4, db_path + 'RBM\\encode\\rbm_model.ckpt')

test_cost = model_rbm.getFreeEnergy(df_x_rbm).reshape(-1)

df_x_lstm = StandardScaler().fit_transform(df_x_lstm)
df_x_lstm = df_x_lstm.reshape(df_x_lstm.shape[0], 1, df_x_lstm.shape[1])

# Load model
model = load_model(db_path + 'Lstm\\encode\\Lstm_Model.h5')

y_predicted = model.predict_classes([df_x_lstm], verbose=0)
# reduce to 1d array
#yhat_probs = yhat_probs[:, 0]
y_predicted = y_predicted[:,0]

counter = Counter(y_predicted)
print(counter)

```

Şekil 4.18. LSTM ve RBM modelin çağrı ve tahminde bulunmaları Python kodu.

Yine aynı veri set yani tamamı, ama otomatik kodlayıcı model'inden geçirerek 15 sütuna düşürmekte olup, İkinci final veri set'imiz olarak, RBM ve LSTM modellerin İkinci veri set'in üzerinde eğitilmiş olanı çağırarak, bagging topluluk sınıflandırıcı model'inde, her işlem için tahminde bulunmuştur. (Şekil 4.19).

```

model1_predict = y_predicted
model2_predict = temp_dt #test_cost

# create a boolean array to check if any prediction is greater than 0
is_positive = np.logical_or(model1_predict > 0, model2_predict > 0)

# create a final prediction based on the boolean array
final_predict = np.where(is_positive, 1, 0)

cm = confusion_matrix(y_true=df_y_lstm, y_pred=final_predict)
print(cm)

fpr, tpr, _ = roc_curve(df_y_lstm, final_predict)
fpr_micro, tpr_micro, _ = roc_curve(df_y_lstm, final_predict)
roc_auc = auc(fpr_micro, tpr_micro)
print('ROC curve (area = %0.2f)' % roc_auc)

plt.plot(fpr, tpr, color='darkorange',
         lw=2, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC curve on val data set')
plt.legend(loc="lower right")
plt.show()

```

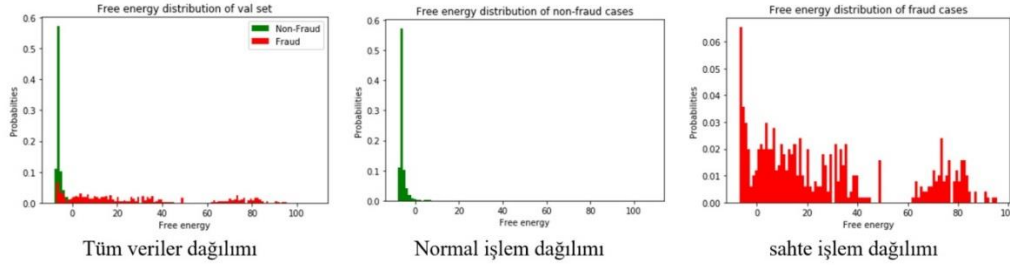
Şekil 4.19. Topluluk sınıflandırıcı modelin Python kodu.

5. DENEYSEL BULGULAR

Bu tez çalışmasında, kredi kartı dolandırıcılık tespiti için RBM, LSTM ve Bagging topluluk sınıflandırıcı model olarak, üç yöntem geliştirilmiştir. Modellerin çıktılarını karşılaştırırken, çeşitli farklı performans değerlendirme ölçümü alınmıştır. Bunlar, AUC-ROC, AUC-PR, kesinlik, geri çağırma, F1-ölçütü, karışıklık matrisi ve hız (çalışma süresi) içeren sınıflandırma performanslarıdır. Bu değerlendirme metrikleri, her üç farklı yöntem için hesaplanmıştır ve her yöntem ayrıca, diğer araştırmalar ile kıyaslanmış olup yedince bölümünde sonuçlanmıştır.

5.1 RBM Modelinde Elde Edilen DeneySEL Bulgular

Şekil 5.1’de, soldaki grafik, normal ve sahte işlem RBM üzerinden geçirilen dağılımını gösterirken, diğer iki grafik ayrı ayrı işlem dağılımını göstermektedir. Yeşil renkle belirlenen, normal işlemin dağılımını yansıtırken, kırmızı renkli olanlar sahte işlemin dağılımını göstermektedir. Görüldüğü gibi, Normal işlemlerdeki tutumun hemen hemen aynı olduğu ancak sahte işlemlerde çok yönlü olduğu anlayabiliriz.

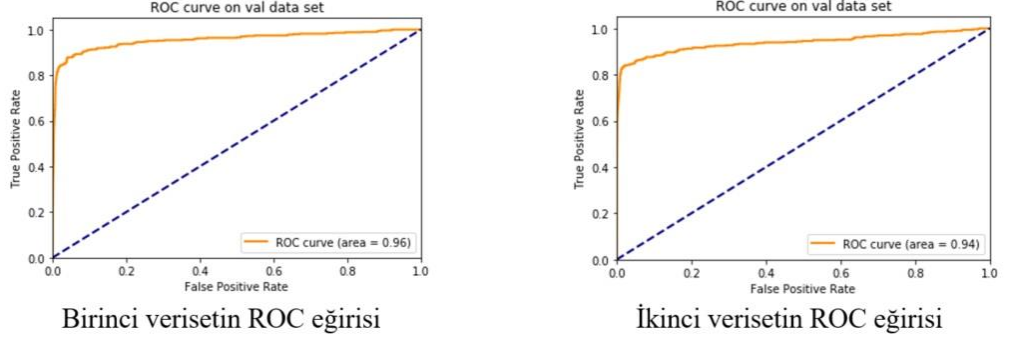


Şekil 5.1. Normal ve dolandırıcılık işlemlerin RBM model ile dağılımı.

Makine öğrenme modelinin oluşturup çalıştırmaktan ziyade en önemli kısmı, onun değerlendirmesidir. Daha önce de söylediğimiz gibi, araştırmamızda aşağıdaki değerlendirme kriterler kontrol edilmiştir:

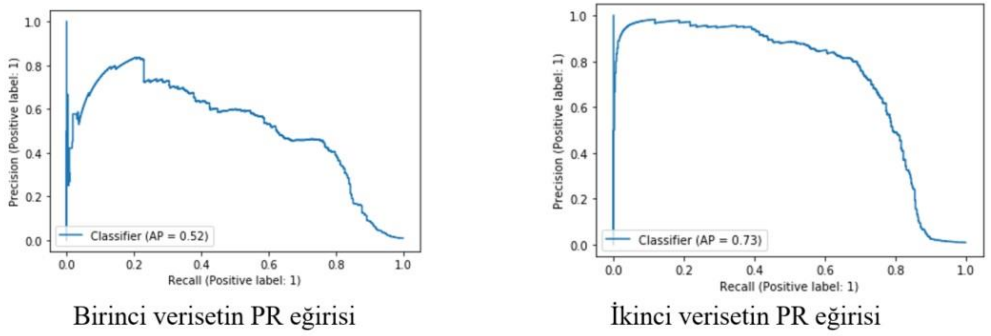
AUC-ROC: sınıflandırıcımızın farklı sınıfları ne kadar iyi ayırt ettiğini gösteren bir parametredir. ROC değeri ne kadar yüksek olursa, model’in dolandırıcılık işlemlerini dolandırıcılık olarak ve normal işlemleri normal olarak tahmin etme kapasitesi o kadar iyi olduğunu göstermektedir. Bir ROC eğrisi iki parametre çizer: True Positive Rate ve False Positive Rate. Şekil 5.2’de gördüğümüz gibi, her iki veri seti için, hemen hemen aynı ROC değerler elde

edilmiştir, ancak birinci veri setinde 2% farkla, ikinci veri sete göre daha iyi sonuç alındığını görebiliyoruz.



Şekil 5.2. RBM ile elde edilen birinci ve ikinci veri setlerin ROC eğrileri.

AUC-PR: Bu parametre PR eğrisini veya başka bir deyişle matematiksel integral altındaki alanı göstermektedir. PR eğrisi, her bir geri çağırma miktarında sınıflandırıcının karşılık gelen kesinliğini gösterir. Böylece, PR eğrisi, farklı eşikler için kesinlik ve geri çağırma arasındaki dengeyi sunar. Bu metrik, yüksek sınıf dengesizliği olan ortamlarda çok uygulanabilir. Şekil 5.3'te soldaki birinci veri setinde PR değeri, %52 oluşmuşken, ikinci veri setinde %73 elde edilmiştir. Bu şekilde dikkat ederek, ikinci veri setindeki sıkıştırma, daha önemli verilerin kullanılmasına neden olduğunu, veya çeldirici ve gereksiz verileri atlayarak daha iyi model oluşturmaya yol açıp, birinci veri sete göre daha güçlü sonuçlara yol açtığını söyleyebiliriz.



Şekil 5.3. RBM ile elde edilen birinci ve ikinci veri setlerin PR eğrileri.

F1-ölçüt: Kesinlik ve geri çağırmanın arasındaki dengeye bulan F1-ölçüt metriği olarak tanımlanmaktadır. Değerlendirme metriğin sonuçları, her iki veri seti için çizelge 5.1'de sunulmuştur. Elde edilen sonuçlar, birinci veri seti için, 100 tekrar veya başka bir deyişle epoch yapılırken, ikinci veri setinde 30 tekrar ile eğitilmiştir. Yani süre açısından da birinci veri sete göre, daha az zaman harcarken, aynı anda performans olarak da daha iyi olduğunu görebiliriz.

Çizelge 5.1. RBM’ın birinci ve ikinci veri setin kesinlik, geri çağırma ve f1-ölçüt sonuçları.

Veri set	Kesinlik	Geri çağırma	F1-ölçüt
1	0.57	0.58	0.57
2	0.74	0.69	0.72

Yukarıdaki bahse geçen kavramların tanımı aşağıdaki gibidir:

Geri çağırma: doğru tespit edilen dolandırıcılık işlemlerinin oranını tahmin etmektedir, yani bütün dolandırıcılık işlemlerin, ne kadarını doğru tahmin ettiğini anlatır:

$$Geri\ çağırma = \frac{TP}{TP + FN} \quad (5.1)$$

Kesinlik: doğru bir şekilde tespit edilip rapor edilen dolandırıcılıkların yüzdelere göstermektedir, yani doğru diye tahmin ettiği işlemlerin ne kadarını doğru tahmin ettiğini anlatır:

$$Kesinlik = \frac{TP}{TP + FP} \quad (5.2)$$

F1-ölçüt: Kesinlik ve geri çağırmanın harmonik ortalamasıdır:

$$F1 - ölçüt = 2 \times \frac{kesinlik \times geri\ çağırma}{kesinlik + geri\ çağırma} \quad (5.3)$$

Karışıklık matrisi: Bu matriste TP, hileli olarak tahmin edilen ve gerçekten hileli olan işlemlerin sayısıdır. FP, hileli olarak tahmin edilen ama normal olan işlemlerin sayısıdır. FN, normal olarak tahmin edilen ama hileli olan işlemlerin sayısıdır. TN, normal olarak tahmin edilen ve gerçekten normal olan işlemlerin sayısını gösterir. Çizelge 5.2’de görüldüğü gibi, bütün karışıklık matrisin parametrelerini kıyaslayınca, ikinci veri seti, birinci veri sete göre daha iyi performans sergilediğini anlayabilmekteyiz. Özetlemek gerekirse, ikinci veri setinde, AE'den geçirilmiş ve boyut indirgeme yapılan veri setinde, iyi performans gösterirken, zaman kazanmayı da desteklenmektedir.

Çizelge 5.2. RBM'in birinci ve ikinci veri setin karışıklık matrisin sonuçları.

Veri set	TP	%TP	TN	%TN	FP	FN
1	285	%58	56253	%99.61	207	217
2	340	%69	56353	%99.79	152	117

Time: Son değerlendirme metriği olarak model'in çalışma süresi saniye bazında getirilmiştir. Ancak bu metrik, çalıştığı sistem'in konfigine göre değişebilmektedir. Bu yüzden çalıştığımız sistemin konfige aşağıda açıklanmıştır:

- İşletim Sistemi: windows 10 Enterprise-64 bit
- CPU: Intel Core (TM) i7-4700 HQ – 2.4 GHz
- RAM: 8 GB DDR3
- Graphic: Geforce GTX 760 M – 2 GB VRAM
- Hard: 1 TB SATA

Çizelge 5.3'e göre Birinci veri setin daha fazla özniteliğe sahip olduğundan dolayı ikinci veri set'e göre daha yavaş olduğunu görebilmekteyiz. Başka bir deyişle öznitelik çıkramanın, süre açısından faydalı olduğunun farkına varabiliriz ve aynı anda daha iyi performans sergilediğini söyleyebiliriz.

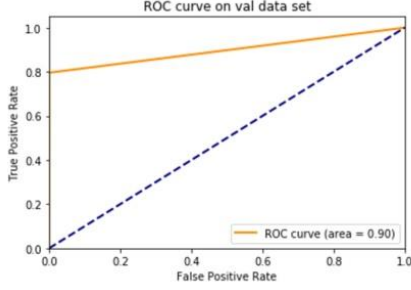
Çizelge 5.3. RBM'in işlem süresi sonuçları.

Veri set	Süre (saniye)
1	365,67
2	65,19

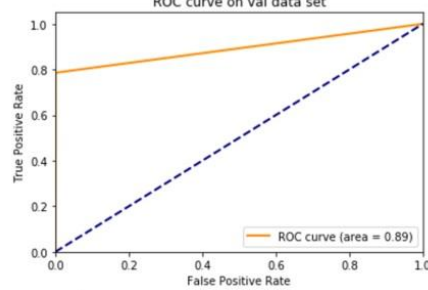
5.2. LSTM Modelinde Elde Edilen Deneysel Bulgular

Daha önce de belirttiğimiz ve kavramlarını açıkladığımız gibi, araştırmamızda aşağıdaki değerlendirme kriterler kontrol edilip burada sadece sonuçlar açıklanmıştır:

Şekil 5.4'te gördüğümüz gibi, birinci veri set için %90 ve ikinci veri set ise %89 ROC değeri elde edilmiştir. Bu sonuç RBM modelimizdeki gibi, her iki veri setinde hemen hemen aynı değeri elde etmiştir, ama RBM modelinde en iyi olan %96'iken LSTM modelinde %90 kayıt edilmiştir.



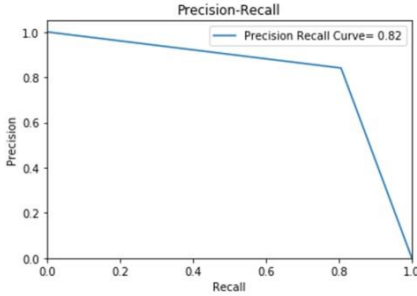
Birinci verisetin ROC eğrisi



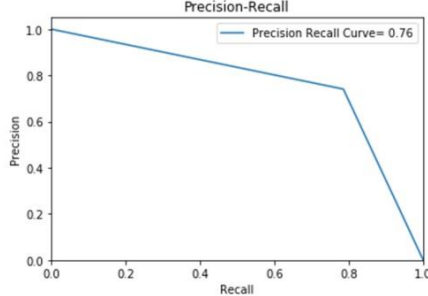
İkinci verisetin ROC eğrisi

Şekil 5.4. LSTM ile elde edilen birinci ve ikinci veri setlerin ROC eğrileri.

Şekil 5.5'te, birinci veri set AUC-PR değeri %82 gösterirken, ikinci veri set için %76 kayıt edilmiştir. Bu metrikte birinci veri seti, ikinci veri setine göre %6 farkla daha iyi olduğunu anlayabiliriz.



Birinci verisetin PR eğrisi



İkinci verisetin PR eğrisi

Şekil 5.5. LSTM ile elde edilen birinci ve ikinci veri setlerin PR eğrileri.

Kesinlik, geri çağırma ve F1-ölçüt sonuçları çizelge 5.4'te getirilmiştir. Bu çizelgeye göre, LSTM modelimizde, her iki veri setinde çok yakın bir performans göstermektedir. Biraz yukarıda da belirttiğimiz gibi, birinci veri setinde %6 fark ile kesinlik metriği iyi olduğundan dolayı, F1-ölçütü de daha yüksek kayıt edilmiştir.

Çizelge 5.4. LSTM'ın birinci ve ikinci veri setin kesinlik, geri çağırma ve f1-ölçüt sonuçları.

Veri set	Kesinlik	Geri çağırma	F1-ölçüt
1	0.84	0.81	0.82
2	0.74	0.79	0.76

Karışıklık matrisin sonuçları çizelge 5.5'te getirilmiştir. Bu sonuçlara göre, önceki incelenen metrikler gibi çok yakın performans sergilediğini görebilmekteyiz. FP ve FN açısından çok iyi olup en düşük seviyede kayıt edildiğini söyleyebiliriz.

Çizelge 5.5. LSTM'ın birinci ve ikinci veri setin karışıklık matrisin sonuçları.

Veri set	TP	%TP	TN	%TN	FP	FN
1	79	%81	56849	%99.97	19	15
2	77	%79	56837	%99.95	21	27

LSTM ile RBM modelin değerlendirmelerini kıyaslayınca, LSTM'ın daha iyi performans sergilediğini fark edebiliyoruz. Bunun en önemli sebepleri, veri setimizde iki günlük işlemlerde pek yeni veya daha önce bilinmeyen dolandırıcılık işlemin az olması ve çok sayıda sağlam işlem olması diye söyleyebilir. Diğer taraftan RBM modelimiz normal dağılımından hariç işlemleri iyi algılayıp, yani yeni tip veya daha önce bilinmeyen dolandırıcılık işlemleri tahmin konusunda, avantaj sağlamaktadır. Bu iki yöntem avantajlarını kullanabilmek adına topluluk sınıflandırıcı model geliştirme ve kullanmaya planlayıp elde edilen deneysel bulgular 5.3 bölümünde açıklanmıştır.

Daha önce açıkladığımız laptop konfig'e göre LSTM'ın işlem süresi performansı Çizelge 5.6'da getirilmiştir. Bu çizelgeye göre ikinci veri setin performansı birinci veri set'e göre, neredeyse 1/3 hızlı çalıştığını görebiliyoruz. Bu araştırmanın amacına göre, ikinci veri setinde, birinciye göre çok benzer performans gösterirken, zaman açısından çok iyi fark kazanmamıza yol açtığını yorumlayabiliriz.

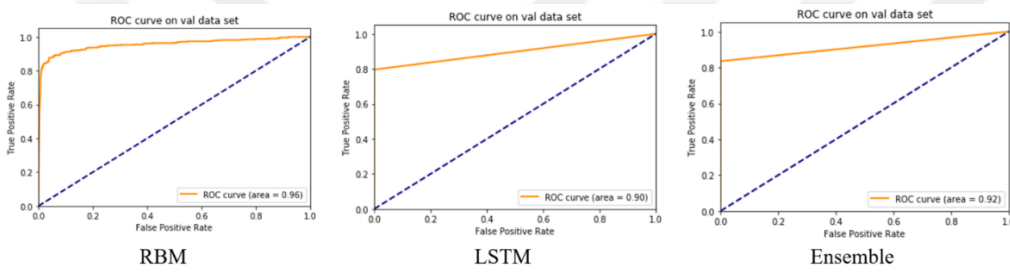
Çizelge 5.6. LSTM'ın işlem süresi sonuçları.

Veri set	Süre (saniye)
1	299,82
2	212,12

5.3 Topluluk Sınıflandırıcı ile Elde Edilen DeneySEL Bulgular

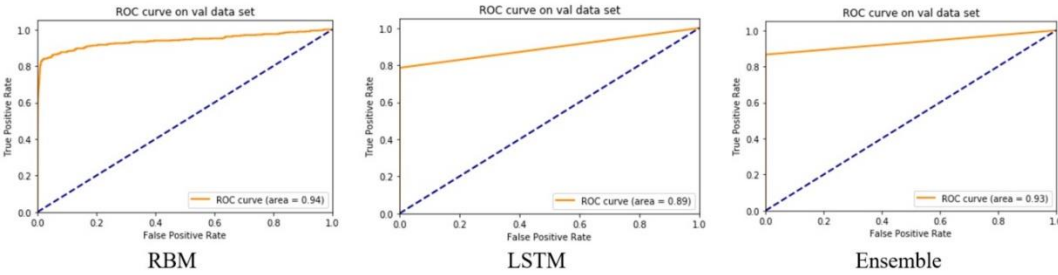
Tüm değerlendirme metriklerini daha önce açıkladığımıza göre, çalışmanın bu bölümünde, her bir değerlendirme metriği için topluluk sınıflandırıcı modelin sonuçlarını RBM ve LSTM modelleriyle kıyaslamaktayız.

İlk değerlendirme ölçütümüz olarak, Şekil 5.6'da ROC eğrisinde karşılaştırma sonuçlarını göstermekteyiz. Görüldüğü gibi, Birinci veri set'imizde, RBM model'inden %96 elde etmişken, LSTM için %90 elde etmiş olup, topluluk sınıflandırıcı modelinde birinci final veri setinde %92 ROC değeri elde edilmiştir. Özet olarak, topluluk modelimiz iki performansın arasını bulmuş diyebiliriz.



Şekil 5.6. Birinci veri setin, RBM, LSTM ve topluluk modellerin AUC-ROC eğrileri.

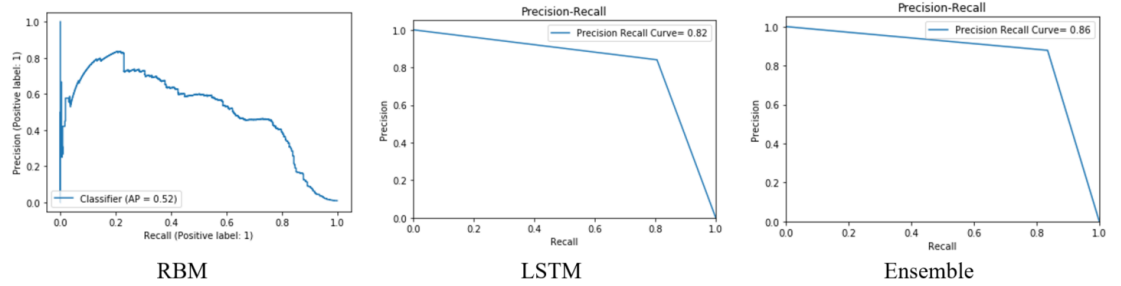
Aynı parametrenin değeri, İkinci final veri set için Şekil 5.7'de göstermekteyiz. Bu değer RBM modeli için %94'iken ve LSTM için %89'iken, topluluk sınıflandırıcı model için %93 elde edilmiştir. Bu sonuçlara göre, bu sıkıştırılmış veri set, topluluk sınıflandırıcı modeli için hiç bilgi kaybına yol açmamış olup, önemli verilerin korumasını sağladığını göstermektedir.



Şekil 5.7. İkinci veri setin, RBM, LSTM ve topluluk modellerin AUC-ROC eğrileri.

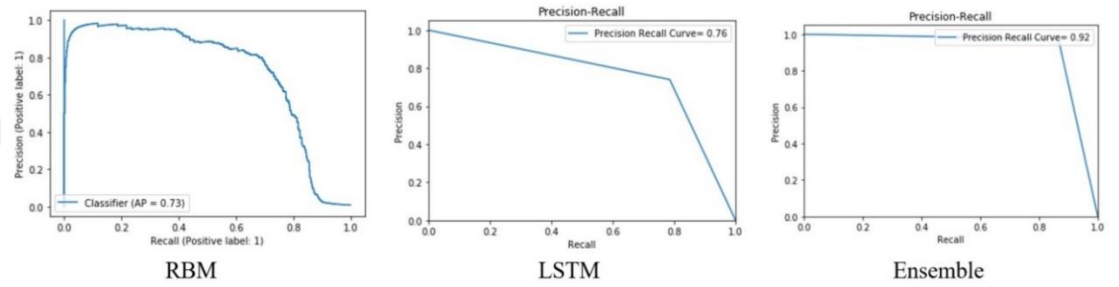
Diğer parametre, PR eğri olarak, Birinci final veri set için Şekil 5.8'de sunulmuştur. Bagging topluluk sınıflandırıcı modelimizde bu değer %86 olarak, her iki solo modelden daha da yüksek seviyeye ulaşmıştır. Bu değerler çok

sevindirici olup, sonraki parametrelerinde de daha iyi olacağının anlamına gelmektedir.



Şekil 5.8. Birinci veri setin, RBM, LSTM ve topluluk modellerin AUC-PR eğrileri.

Aynı parametre İkinci final veri seti için, şekil 5.9’da gösterilmektedir. İkinci final veri setimizin PR değeri %92 kayıt olup, diğer iki solo modellerden daha iyi olduğunu ve hatta birinci final veri setinden de daha iyi olduğunu anlayabilmekteyiz.



Şekil 5.9. İkinci veri setin, RBM, LSTM ve topluluk modellerin AUC-PR eğrileri.

Kesinlik, geri çağırma ve F1-ölçüt parametrelerin kıyaslaması, Birinci final veri set için Çizelge 5.7’de getirilmiştir. çizelgeye göre, Birinci final veri set’imiz topluluk modelimizde, her iki solo modele göre daha iyi olduğunu ve sonra LSTM modelimiz az farkla iyi olduğunu görebilmekteyiz. Burada RBM modelinde az da olsa LSTM modelin tahmin edemediği bazı dolandırıcılık işlemlerin, doğru tahminde bulunarak, topluluk için verimli olduğunu söyleyebiliriz.

Çizelge 5.7. Birinci veri setin modeller arası kesinlik, geri çağırma ve f1-ölçüt sonuçları.

Model	Kesinlik	Geri çağırma	F1-ölçüt
Topluluk	0.88	0.84	0.86
RBM	0.57	0.58	0.57
LSTM	0.84	0.81	0.82

Çizelge 5.8’de İkinci final veri setin aynı değerlendirme parametreler sunulmuştur. İkinci final veri setinde, topluluk modelimizde çok iyi sonuçlar elde edildiğini ve daha doğrusu, solo RBM ve LSTM modelimizde birbirlerinin tahminlerini ne kadar güzel örttüğünü ve tamamladığını görebilmekteyiz. Şimdiye kadar ikinci veri setimiz topluluk modelimizin, sadece diğer iki solo modelden değil belki birinci veri setimizin topluluk modeline göre de daha iyi olduğunu yorumlayabiliriz.

Çizelge 5.8. İkinci veri setin modeller arası kesinlik, geri çağırma ve f1-ölçüt sonuçları.

Model	Kesinlik	Geri çağırma	F1-ölçüt
Topluluk	0.97	0.87	0.92
RBM	0.74	0.69	0.72
LSTM	0.74	0.79	0.76

Karışıklık matris metriğin karşılaştıması Birinci veri seti için çizelge 5.9’da getirilmiştir. Burada topluluk sınıflandırıcı modelimizin ne kadar iyi güçlenmiş olup, iyi tahmin ettiğini görebiliriz. TP açısından, bahse geçen topluluk sınıflandırıcı model 492 mevcut dolandırıcılık işlemde 411 tanesini doğru tahmin etmiş yani %84 doğru. TN açısından %99.97 doğru işlemi, doğru olarak tanımlamış. FP ve FN olarak oran bazlara bakınca, en düşük yanlış elde etmiştir. Böylece, topluluk sınıflandırıcı modelimizin, iki modelin avantajlarını bir araya getirip, ne kadar iyi performans sergilediğini kanıtlayabiliriz.

Çizelge 5.9. Birinci veri setin karışıklık matrisin sonuçları.

Model	TP	%TP	TN	%TN	FP	%FP	FN	%FN
Topluluk	411	%84	284258	%99.97	81	%16	57	2.0048e-4
RBM	285	%58	56253	%99.61	207	%42	217	0.0038
LSTM	79	%81	56849	%99.97	19	%29	15	3.3413e-4

İkinci final veri set’in karışıklık matrisin kıyaslaması çizelge 5.10’da görüntülenmektedir. Daha öncede belirttiğimiz gibi, bu sıkıştırılmış veri setinde, topluluk sınıflandırıcı metodu için çok ideal olup, bütün parametrelerde en iyi sonuçlar elde edildiğini görebiliyoruz.

Çizelge 5.10. İkinci veri setin karışıklık matrisin sonuçları.

Model	TP	%TP	TN	%TN	FP	%FP	FN	%FN
Topluluk	426	%86	284304	%99.99	66	%14	11	3.8689e-5
RBM	340	%69	56353	%99.79	152	%31	117	0.0020
LSTM	77	%79	56837	%99.95	21	%21	27	4.7482e-4

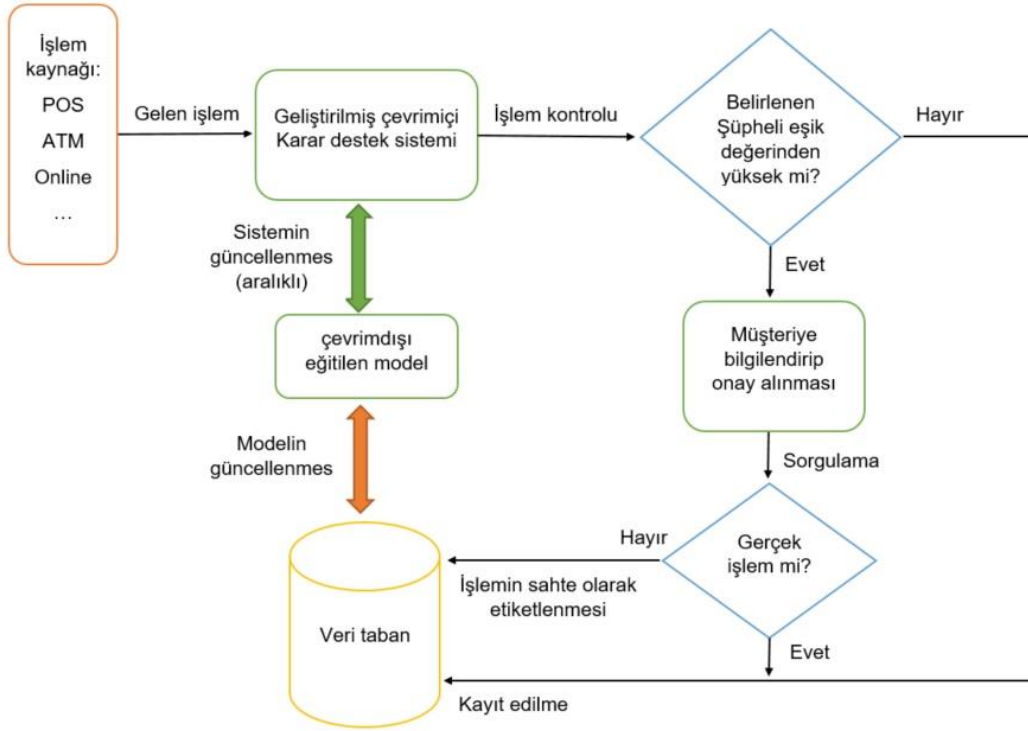
Son metriğimiz olarak, en iyi sonucumuz, ikinci veri setimizin topluluk sınıflandırıcı modelinde çalıştırıp tahminde bulunması '34,40' saniye sürmüştür. Bu '286.807' işlemler için gayit uygun bir işlem süresi olduğunu söyleyebiliriz.

Çizelge 5.11. Topluluğun işlem süresi sonuçları.

Veri set	Süre (saniye)
1	49,33
2	34,40

6. GELİŞTİRİLEN KARAR DESTEK SİSTEMİNİN ÇALIŞMA PRENSİBİ

Bu tez çalışması kapsamında, kredi kartı dolandırıcılık tespiti için gerçekleştirilen karar destek sisteminin performans ve sonuçlarına göre mümkün olduğu anlaşılmıştır. Buna bağlı olarak Şekil 6.1’de geliştirilmiş karar destek sisteminin çalışma prensibi sunulmuştur.



Şekil 6.1. Geliştirilen karar destek sisteminin çalışma prensibi.

Şematiğe göre, tasarlanan modelde bir işlem isteği, hangi araç yoluyla servere girdikten sonra, bu işlem geliştirilmiş çevrimiçi karar destek sistemine girmiş olmaktadır. Bu sistemde girdikten sonra, bu işlemin dolandırıcılık seviyesini veya başka bir deyişle şüpheli seviyesini, sistemimizde belirlenen eşik ile kıyaslayıp, eğer bu seviyeyi aşmazsa, veri tabanda kayıt olup işlem bitmiş olmaktadır. Eşik değerin aşılması durumunda, müşteri bilgilendirip sorgulama yapılmaktadır. Bu işlem gerçek ise, onu sağlam işlem olarak etiketleyip veri tabanda kayıt ediyoruz, ama sahte ise sahte etiket yapıştırıp kayıt ediyoruz. Bu yeni kayıt olan işlemler ile, çevrimdışı eğitilmiş modelimizi güncelleyip, belirli aralıklarla çevrimiçi karar destek sistemi de güncellenecektir.

Tasarlamış olduğumuz modelde şüpheli olan işlemlerin sorgulanması ve müşteriden onay alınması kısmında bir sms veya yazılım yoluyla

gerçekleştirerek, eksper ve alıřanı ortadan kaldırmıř olacak ve bylece tamamen otomatik karar destek sistemi gerekleřtirilebilecektir. Eksper ortadan kaldırılması, insani hataları engellerken, bir yandan gece ve uygun olmayan saatlerde bir kiřinin gzlenmesine gerek kalmamıř olacaktır. Dięer taraftan hata seviyesi en dřk miktara indirilmiř olacaktır.



7. TARTIŞMA

Bu başlık altında, tez çalışmasında gerçekleştirilen deneyler ile literatürde yer alan diğer gerçekleştirilen kredi kartı dolandırıcılık tespiti çalışmaları ile karşılaştırılacaktır. Bu karşılaştırmalar hem solo modeller hem de topluluk modelleri ile ilgili çalışmalarla gerçekleştirilmiştir.

7.1 RBM sonuçlarını diğer araştırmalar ile Karşılaştırma

Solo RBM modelinde elde edilen deneysel bulguları başka makaleler ile karşılaştırma Çizelge 7.1’de getirilmiştir. AE ve RBM benzer yapılara sahip olduğundan dolayı, AE olan araştırmaların sonuçları da dahil edilmiştir, ancak dikkat edilirse değerlendirme metrikleri kısmında, bizim parametreler ile ortak olanları yansıtılmıştır ve tire ile dolduranlar, o araştırmalarda performans olarak kullanılmamış anlamına gelmektedir.

Çizelge 7.1. RBM’in elde edilen sonuçların diğer araştırmalar ile karşılaştırma.

Model	Veri set	AUC-ROC	AUC-PR	Kesinlik	Geri çağırma	F1-ölçüt	Süre
Geliştirilen RBM	ECH 1. veri set	%96	%52	%57	%58	%57	365.67
	ECH 2. veri set	%94	%73	%74	%69	%72	65.19
AE ve RBM	USD veris set	-	-	%71	-	-	-
	Pysim simüle	-	-	%57	-	-	-
	Banka veris set	-	-	%45	-	-	-
AE ve derin NN	ECH	-	-	%19	%58	%29	-
AE	ECH	%96	-	-	-	-	-
RBM	ECH	%95	-	-	-	-	-
AE	ECH	%96	-	-	-	-	-
Variational AE	ECH	%96	-	-	-	-	-
RBM	ECH	%96	-	-	-	-	-
AE	ECH	%89	-	-	-	-	-
RBM	ECH	%91	-	-	-	-	-

Çizelgenin ikinci satırında, Bagchi et al. (2021), araştırmalarında AE ve RBM olarak belirlenen modelini aşağıdaki üç veri kümesi üzerinde incelemişler.

1) California Üniversitesi, San Diego (UCSD) veri seti, 100.000 kredi kartı işlemi kaydını içeren ve her kaydın 20 özelliğe sahip olan gerçek bir e-ticaret işlemlerin veri seti.

2) Dolandırıcılık Tespiti İçin Sentetik Finansal Veri Kümesi: PaySim mobil para simülatörü tarafından oluşturulan veri kümesi.

3) Dolandırıcılık tespiti için banka veri seti: müşterinin 20k işlemler ve 112 özellikli bir veri kümesi. Değerlendirme metrikleri doğruluk, kesinlik ve yanlış

alarm oranı şeklinde yapılmış. Ortak olan performanslarını, geliştirilen RBM modelimizin sonuçları ile kıyaslayınca, Veri kümeleri farklı olsa da, RBM modelimizde süreyi dikkate alınmasına rağmen daha iyi performans sergilediğini görebilmekteyiz.

Yukarıdaki çizelgenin üçüncü satırında Abakarim et al. (2018) ECH veri setini kullanırken yaptıkları araştırmada Derin yapay sinir Otomatik Kodlayıcıyı önermişler. Çalışmalarındaki bazı denetimli ve denetimsiz ML modelleriyle karşılaştırıldığında önerilen Derin yapay sinir otomatik kodlayıcı modelde en yüksek değeri elde etmişler. Geliştirilen RBM modelimizle karşılaştırıldığında, sonuçlarımız büyük bir farkla daha iyi olduğu görülecektir.

Yukarıdaki tablonun 4. satırda, Pumsirirat et al (2018) otomatik kodlayıcı ve RBM modellerini incelerken Avrupa, Almanya ve Avustralya dahil üç farklı kredi kartı veri seti göz önünde bulundurarak değerlendirme ölçütleri için AUC-ROC ve karışıklık matrisini kullanmışlar. Otomatik kodlayıcı ve RBM'e odaklanan makalelerden birisi olmasına dikkat ederek, çalışmalarında hassasiyet, hatırlama ve F1 ölçümü gibi bazı önemli metriklerden yoksun kalmaları, eksiklik sayılacaktır.

Beşinci sırada Reshma et al. (2018) ECH veri setini kullanarak otomatik kodlayıcı, RBM ve Variational RBM gibi farklı Derin İnanç Ağı tabanlı modeller arasında bir karşılaştırma yapmışlar. Çalışmaları, değerlendirme ölçümleri için AUC-ROC ve karışıklık matrisini kullanırken, daha önce açıklanan bazı önemli ölçümlerden de yoksunluğu, araştırmalarında eksik olduğunu söyleyebiliriz.

Çizelgenin son satırında Raghavan et al. (2019) Avrupa, Almanya ve Avustralya olmak üzere üç farklı kredi kartı veri setini kullanarak derin öğrenme ve makine öğrenme tabanlı modeller arasında karşılaştırma sonuçlar verilmektedir. Çalışmalarının kör noktası, sadece Matthew Korelasyon Katsayısı (MCC), AUC-ROC ve başarısızlık maliyetinden oluşan, kullanılan değerlendirme ölçüleridir.

Yukarıda incelenen RBM modelleriyle ilgili makalelerin sonuçları dikkat edilerek, önerdiğimiz RBM modelimiz, özellikle öznitelik çıkarılmış modelimiz, diğer incelenen modeller arasında en iyi sonuç ve tam değerlendirmeye sahip olduğunu ve ayrıca gerçek zamanlı tespit için harcama süresi metriği de dikkate alınmış olduğunu söyleyebiliriz.

7.2 LSTM sonuçları diğer araştırmalar ile karşılaştırma

Bu bölümde solo RBM bölümündeki gibi, solo LSTM modelimizde elde edilen deneysel bulguların, diğer araştırmalardaki LSTM modelleri ile karşılaştırma Çizelge 7.2’de getirilmiştir.

Çizelge 7.2. LSTM’ın elde edilen sonuçların diğer araştırmalar ile karşılaştırma.

Model	Veri set	AUC-ROC	AUC-PR	Kesinlik	Geri çağırma	F1-ölçüt	Süre
Geliştirilen LSTM	ECH 1. veri set	%90	%82	%84	%81	%82	299.82
	ECH 2. veri set	%89	%76	%74	%79	%76	212.12
Topluluk LSTM 2li	ECH	%86	%67	%92	%73	%81	569.31
Topluluk LSTM 3lü	ECH	%86	%68	%93	%73	%82	749.39
Topluluk LSTM 4lü	ECH	%83	%64	%97	%66	%78	919.34
Topluluk LSTM 5li	ECH	%83	%64	%96	%66	%78	1141.13
DNN	IEEE - CIS	%84	-	-	-	-	-
LSTM without resampling	Custom MC	-	-	%61	%9	%71	-
LSTM with overampling	Custom MC	-	-	%63	%7	%71	-
LSTM with undersampling	Custom MC	-	-	%33	%4	%42	-

Forough et al. (2021) yukarıdaki çizelgenin ikinci satırında RNN tabanlı modeller üzerine bir çalışma önermişlerdir. GRU ve LSTM'yi Avrupa ve Brezilya dahil iki farklı kredi kartı veri kümesinde karşılaştırmışlar. Modelleri eğittikten sonra, bunları önerilen topluluk modelinde İleri Beslemeli yapay sinir ve Çoğunluk Oylaması için temel sınıflandırıcı olarak kullanılmıştır. Her ne kadar topluluk modellerini kullanarak modellerini geliştirmiş olsalar da AUC-PR için kayıtlı değerlerde bazı yanlış anlaşılmalarda bulunmaktadır. Tüm değerler 64 ile 68 aralığındadır, ancak tüm hassasiyet, geri çağırma ve özellikle F1-ölçütü değerleri daha yüksek kayıt edilmiştir, halbuki F1-ölçütü miktarı AUC-PR'nin kayıtlı aralığı civarında ve hatta altında olması gerekmektedir. Ama bu çalışma, değerlendirme parametreleri açısından en nadir tam makalelerden birisi olduğunu söylememiz gerekir. Bizim modellerimiz ile kıyaslayınca bizim birinci veri setimiz ile çok yakın performansta olduğunu ancak onlar topluluk model ile geliştirirken, bu çalışmada solo LSTM model performansı getirilmiştir.

Üçüncü sırada, Nghia Nguyen et al. (2021), IEEE-CIS veri setin üzerinde derin yapay sinir modeli geliştirip değerlendirmiştir. Kendi modelinde catboost ile karşılaştırma, burada sadece yapay sinir modelinin yapısı LTSM ile benzetmekten dolayı, tabloda performansı getirilmiştir. diğer parametreleri, birisi doğruluk olup diğerleri AUC-ROC, AUC-PR şekilde bir değer verilmeden

sadece eğim şeklini sunmuşlar. Elbette, sonuçlara ve değerlendirme metriklerine bakılarak, bu makalenin biraz daha geliştirilebileceğini söylemek mümkündür.

Wu et al. (2023), dengesiz veri setleri üzerinde çalışmak için aşırı örnekleme ve yetersiz örnekleme tekniklerini kullanmışlardır, bu durumu tablonun son satırında açıklanmıştır. Aşırı örnekleme, yetersiz örnekleme ve yeniden örnekleme olmadan 7.352.421 işlem içeren özel bir Mastercard veri seti üzerinde araştırma yapmışlardır. LSTM'yi, kendi modelleri içerisinde neredeyse en iyi sonucu elde edip, Doğrusal Regresyon ve Rastgele Orman modelleriyle karşılaştırmışlardır. Bu çalışmada da bazı yanlış anlaşılma değerleri yer almakta; F1-ölçüt metriği, kesinlik ve geri çağırma değerleri arasındaki ortalamasını döndürür, ancak F1-ölçüt'te en yüksek değerleri kaydederken, geri çağırma için tüm değerler %10'un altında kayıt edilmiştir.

Yukarıdaki LSTM bazlı modeller inceledikten sonra özetlemek gerekirse, sadece ikinci sıradaki çalışmada çok detaylı bir inceleme yapmış olmuş ancak yukarıda da açıkladığımız gibi bazı yanlış anlaşılma mevcutmuş, bu yüzden geliştirilen SOLO LSTM modelimiz tam ve hatasız bir araştırma olduğunu söyleyebiliriz.

7.3 Tez Kapsamında Elde Edilen Bulguların Karşılaştırılması

Araştırmış olduğumuz son güncel makalelerde, yapay zeka ürünlerin gündemde kullanışlı olsa bile, daha öncede belirttiğimiz için, çevrimiçi kredi kartı dolandırıcılık tespitine yönelik pek çalışma bulamamaktayız. Bu yüzden kendi çalışmalarımızı, topluluk sınıflandırıcı olan çalışmalar veya çok az sayıda olan çevrimiçi kredi kartı dolandırıcılığa yönelik çalışmalar ile karşılaştırma yapılmış. Aslında çevrimiçiye yönelik değerlendirme parametresi olarak "Hızı" kontrol etsek bile, bu parametrenin, çalıştığımız sistemin konfigine bağlı olduğundan dolayı, başka araştırmalarla karşılaştırma mantıklı gelmemektedir, ancak tasarlandığı karar destek sistemin, gerçek sahada ne kadar uygulanabileceği konusunda yardımcı olmaktadır.

Çizelge 7.3. Topluluk sınıflandırıcı sonuçların diğer araştırmalar ile karşılaştırma.

Model	Veri set	AUC-ROC	AUC-PR	Kesinlik	Geri çağırma	F1-ölçüt	Süre (s)
Geliştirilen Topluluk	ECH 1. veri set	%92	%86	%88	%84	%86	49.33
	ECH 2. veri set	%93	%92	%97	%87	%92	34.30
AE-DNN	ECH	%90	%77	%96	%72	%82	-
AE-RNN	ECH	%89	%76	%97	%72	%83	-
AE-CNN_RNN	ECH	%92	%75	%96	%73	%83	-
AE-SVM	ECH	%86	%71	%97	%72	%83	-
Topluluk LSTM 2li	ECH	%86	%67	%92	%73	%81	87.16

Topluluk LSTM 3lü	ECH	%86	%68	%93	%73	%82	122.43
Topluluk LSTM 4lü	ECH	%83	%64	%97	%66	%78	160.75
Topluluk LSTM 5li	ECH	%83	%64	%96	%66	%78	216.94
Logistic-BWS	ECH	%88	-	-	-	%84	-
logistic	ECH	%84	-	-	-	%73	-
DT	ECH	%85	-	-	-	%82	-
k-NN	ECH	%85	-	-	-	%83	-
SVM-rbf	ECH	%85	-	-	-	%83	-
AdaBoost	ECH	%85	-	-	-	%83	-
XGBoost	ECH	%85	-	-	-	%83	-
Topluluk	ECH	%98.9	-	%99.96	%100	%99.95	-
Pipelining	ECH	-	-	%84	%86	%85	-
Topluluk	ECH	-	-	%79	%87	%81	-
DBDT-COM	ECH	%98.3	-	-	-	-	-

Çizelge 7.3'e göre, ilk sıradaki yer alan bizim araştırma sonuçlarını gösterirken, ikinci sıradaki Fanai et al. (2023), ECH veri setin üzerinde otomatik kodlayıcı modeli çeşitli konfigler ile çalıştırıp, 29 özniteliği 22'e düşürüp en iyi sonuç aldığını kanıtlamışlar. Ancak bu durumda öznitelikler 22'e düşürerek hız açısından iyileştirme yapsa bile, ama araştırmacılar, bunu hiç ele geçirmediğini açıklamamız gerekiyor. performans açısından çok iyi sonuç elde etmiş olsalar da, bizim topluluk modelimizden, özellikle ikinci veri set ile geliştirmiş olandan daha düşük olduğunu kanıtlamaktayız.

Üçüncü sırada Forough et al. (2021) geliştirilen Topluluk LSTM-bazlı modellerin üzerinde araştırmaların sonuçları getirilmiştir. Daha öncede detaylı şekilde açıklandığından dolayı sadece kıyaslamayı açıklayacaksak, ne kadar da tam bir değerlendirme yapmış olsalar bile, performans açısından geliştirdiğimiz modelden daha düşük olduğunu söyleyebiliriz. İşlem süresi açısından, donanım konfigürasyonu ile karşılaştırdınca, onların sistemleri daha güçlü olmasına rağmen, daha yavaş olduğunu açıklayabiliriz.

Dördüncü sırada, Runchi et al. (2023), araştırmalarında Avustralya, Almanya ve Avrupa veri setini kullanmışlar. Logistic-BWE adlı modellerini diğer 10 ünlü sınıflandırma modeliyle karşılaştırılmış olup en iyi olduğunu kanıtlamışlar. Her ne kadar performansları iyi olsa bile, çeşitli parametrelerin eksikliği hissedilmektedir. Onların Logistic-BWE modeli bu çalışmada ikinci veri seti ile geliştirilmiş topluluk modelimizden daha düşük performans sergilediğini görebilmekteyiz.

Beşinci sırada, Islam et al. (2023), ECH veri seti ve CCDDP (Kredi Kartı Varsayılan Ödeme) veri setini kullanmışlar. İlk olarak Karar Ağacı, K en yakın komşu, SVM, OneClass SVM, eEnvelope, iForest ve LOF gibi tekli

sınıflandırıcıların performansını karşılaştırmışlar. Sonra iki katmanlı bir istifleme yöntemi geliştirmişler. İlk katmanda dört denetimsiz öğrenme modelini, ikinci katmanda ise meta sınıflandırıcı olarak XGBoost'u kullanarak kredi kartı anormalliklerinin tespit oranını iyileştirmeyi hedeflemişler. Değerlendirme metriklerinde neredeyse %100'e dikkat edildiğinden sonuçların, overfitting ile elde edilip edilmediği tartışılabilir. Ayrıca modellerinde çok karmaşık sınıflandırıcılar kullanmak, bunları gerçek zamanlı tespitte uygulamak mümkün olmadığını söyleyebiliriz.

Altıncı Sırada, Bagga et al. (2020), Araştırmalarında ECH veri kümesini kullanmışlar. Son sınıflandırıcı tarafından takip edilen bir dizi dönüşümün uygulanması anlamına gelen Pipeling model geliştirmişler. Pipeline, farklı parametrelerin eş zamanlı olarak ayarlanmasıyla birlikte çapraz doğrulama amacıyla birden fazla işlemin bir araya getirilmesi için kullandıklarını ve paketleme topluluğu öğrenme modeli ile sunmuşlardır. Geliştiren modellerini doğrusal regresyon, NB, K en yakın Komşu, Çok Katmanlı Perceptron, Ada Boost ve Rastgele Orman ile karşılaştırmış olup az daha iyileştirdiklerini kanıtlamışlardır.

Yedinci sırada Xu et al. (2023), Araştırmalarında 5 farklı veri seti üzerinde çalışmış olup tabii ki birisini ECH olarak seçmişler. Tüm dolandırıcılık türlerinde en iyi performansı aşan genel bir makine öğrenimi algoritması bulmayı hedeflemişler. DBDT olarak Derin Güçlendirme Karar Ağacı, önerilen modeldi ve temsili öğrenme yeteneğini geliştirmek ve bu arada yorumlanabilirliği korumak için sinir ağlarını gradyan artırmaya yerleştirmişler. Ayrıca, tespit edilen dolandırıcılık vakalarının nadirliğini hedefleyerek, model eğitimi aşamasında, algoritma düzeyindeki veri dengesizlikleriyle başa çıkmak için bileşimsel bir AUC maksimizasyon yaklaşımı önermişler, ancak çalışmalarında çeşitli değerlendirme ölçütlerinin eksikliği görebilmekteyiz.

Tabloya bakıldığında, özellikle ECH 2 ile eğitilmiş olan önerilen topluluğumuzun, beşinci sıra hariç tutulduğunda diğer tüm topluluk modelleri içerisinde en iyi sonuçları aldığı sonucuna varabiliriz. Ayrıca, tüm değerlendirme metriklerinde %100 kazanç elde etmek, beşinci sırada bir şekilde güvensizlik uyandırır ve aşırı uyumun bir sonucu olabilir olmasını açıklamamız gerekiyor. En yüksek hassasiyeti, geri çağırma, F1-ölçüt ve sonuçlarda en kısa süreyi elde etmek, topluluk modelimizin ne kadar başarılı olduğunu söyleyebiliriz.

Çizelge 7.4'te tüm çalışmalardaki karışıklık matrisi karşılaştırması ayrıca sunulmuştur.

Çizelge 7.4. Topluluk sınıflandırıcı karışıklık matrisin sonuçlarını diğer araştırmalar ile kıyası.

Model	TP	%TP	TN	%TN	FP	%FP	FN	%FN
Topluluk1	411	%84	284258	%99.97	81	%16	57	2.0048e-4
Topluluk2	426	%87	284304	%99.99	66	%13	11	3.8689e-5
DNN-AE	358	%20	115793	%99.77	1457	%80	257	0.0060
AE	82	%84	55202	%97.07	16	%16	1662	0.0292
DL-CNN	81	%83	56854	%99.98	17	%17	10	1.7588e-4
AE	80	%82	55480	%97.56	18	%18	1384	0.0243
Variational AE	77	%79	55504	%97.60	21	%21	1360	0.0239
D Belief Network	80	%82	55480	%97.56	18	%18	1384	0.0243

İkinci satırda, Abakarim et al. (2018), daha önce de açıkladığımız gibi Derin Yapay sinir Otomatik Kodlayıcı modellerinde test seti için %40 ECH veri setini kullandıklarını söylemişler. Yani dolandırıcılık sayısının 197 olması gerekirken, toplam dolandırıcılık işlemlerinin ise 1.815 (358 artı 1.457) olduğunu ve bunun anormal olduğunu ifade etmemiz lazım. Her neyse, kayıtlı karışıklık matrislerine dayanarak TP, TN, FP ve FN'in yüzdesi hesaplanmıştır.

Üçüncü sırada Pumsirirat et al. (2018), Otomatik Kodlayıcı modelini geliştirmek için ECH veri kümesine kullanmışlar. Neredeyse iyi sonuçlar elde ettikleri, ancak FN açısından modelleri yeterince iyi çalışmadığını görebiliyoruz.

Dördüncü sırada Alarfaj et al. (2022), Derin Öğrenme CNN bazlı modelinde ve ECH veri seti üzerinde, neredeyse en iyi sonucu elde etmişler olsa bile, bizim topluluk2 model sonuçlarımız kadar iyi olmadığını görebiliyoruz.

Son sırada Reshma et al. (2018), iyi sonuçlar elde etmişler, ancak FN oranı, üçüncü sıra çalışmada anlattığımız gibi yüksek olduğunu fark edebiliyoruz.

Son değerlendirme metriği olarak, daha da önce açıkladığımız gibi, sadece Forough et al. (2021), süre performanslarını kayıt etmişler ve onların çalıştığı sistemin donanımsal olarak bizim çalıştığımız sistemden daha güçlü olmasına rağmen, bizim modelimiz daha hızlı çalıştığını açıklayabiliriz ve sonuçlar çizelge 7.5'te sunulmuştur.

Çizelge 7.5. Modellerin işlem süresi karşılaştırması.

Model	Süre (saniye)
RBM-1	365.67
RBM-2	65.19
LSTM-1	299.82
LSTM-2	212.12
Topluluk-1	49.33
Topluluk-2	34.30
LSTM-2l	482.15
LSTM-3l	626.96
LSTM-4l	758.59
LSTM-5l	924.19
FFNN-2l	87.16
FFNN-3l	122.43
FFNN-4l	160.75
FFNN-5l	216.94

8. SONUÇ

Bu tez çalışmasında kredi kartı dolandırıcılık tespiti için, karar destek sistemi gerçekleştirmek doğrultusunda, Avrupa Kart Sahibi veri setin üzerinde solo RBM, solo LSTM ve torbalama topluluğu modeller geliştirilmiştir. Ancak, yetersiz sayıda veri seti sorunundan dolayı, bu veri setinin üzerinde boyut indirgeme kullanarak, ikinci veri seti oluştururken, bir yandan gerçek zamanlı çalışmamız da desteklenmiştir.

Tez kapsamında boyut indirgeme yöntemlerinden, Otomatik Kodlayıcı modele geliştirerek, boyut indirgeme yöntemin performansı değerlendirilmiştir. Bunun için farklı variance covered parametreleri ile testler yenilenmiştir. Sonunda bu iki veri seti, sırayla hiç dokunmadan birinci veri seti ve boyut indirgeme yapıp ikinci veri seti olarak adlandırıp sonraki modellerimizin geliştirmesinde kullanmak adına hazırlanmıştır.

Birinci model olarak, RBM modeli seçilmiştir. Bunun en önemli sebebi, bu yöntemin gerçek işlemlerin normal dağılımını her zaman koruduğu için yeni sahte işlemleri bulmada çok iyi olduğu özelliğe sahip olmasıdır. Tez çalışmasında geniş kapsamda model konfigleri inceleyip deneme yaparak, her iki veri seti için en iyi konfigler ayarlanıp, modeller, topluluk geliştirmede kullanmak için kayıt edilmiştir.

İkinci modelimiz LSTM olarak seçilmiştir ve en önemli nedeni, bu modelin, işlemlerdeki tüm hileli tutumları hatırlama konusunda çok iyi olmasıdır. Yani tanımlayan bir dolandırıcılık davranışı çok iyi hatırlayıp, benzer gelen dolandırıcılara uyanık olduğunu söyleyebiliriz. Bu çalışmada da RBM'deki gibi geniş araştırma yapıp, en iyi konfigler bulunup, her iki veri setinde modelimiz, topluluk modelde kullanmak adına kayıt edilmiştir.

Topluluk modelimiz, RBM ve LSTM'in bir değiş-tokuş karışımı olarak geliştirilmiştir. Bunların etkisi eşit olarak yani %50 şekilde bir gelen işlemin dolandırıcılık olup olmadığına karar vermek için tasarlanmıştır. Elde edilen sonuçlara göre, modelin öğrenme süresinin azalması amaçlayan veri setinde çok iyi bir performansa yol açtığını ispatlanmıştır. Yani ikinci veri kümesinde etkili olmayan özniteliklerin ayıklanması nihai kararı etkilemeyen bazı önemsiz verileri düşürerek, gelen işlemlere karar vermek için zaman kazanmasına neden olmuştur. Topluluk modelimiz, solo RBM ve LSTM modellerinde en yüksek TP, TN ve en düşük FP ve FN'yi elde ederek en iyi sonuçlar elde edildiği

kanıtlanmıştır. Bu nedenle, bankacılık kayıtlarının gerçek dünyadaki veri kümesinde, bazı gereksiz sütunların sıkıştırılması veya kaldırılması önemle tavsiye edilir.

İki farklı yapıları veri seti ve üç verimli modellerin üzerinde en başarılı sonuç olarak, topluluk modelimizin ikinci veri seti üzerinde olduğunu kayıt ettiğimizi söyleyebiliriz. Sınıflandırıcı topluluklardan max-voting topluluk model yöntemi de öznitelik çıkararak final karar destek sistemi gerçekleştirilmiştir. Karar süresi açısından 284.807 işlemin 34 saniye, iyi bir işlem süresi performansı olarak değerlendirilmiştir. Bu kapsamda literatürde gerçek zamanlı uygulanabilecek bir sistem kazandırılmıştır. Sonuç olarak, hem zaman kazandıran hem de performansı artan bir kredi kartı dolandırıcılık sistem gerçekleştirilmiştir.

Gelecek çalışmalarda, daha fazla gün aralığına sahip farklı gerçek veri kümelerinin bulunması, önerilen mevcut modelin geliştirilmesinde ve optimize edilmesinde ve aynı zamanda finansal kurumlar üzerinde gerçek zamanlı bir kredi kartı dolandırıcılık tespit sisteminin uygulanmasında daha sınıflandırmaların gerçekleşmesine olanak sağlayacaktır.

KAYNAKLAR DİZİNİ

- Abakarim, Y., Lahby, M. and Attioui, A.**, 2018, October. An efficient real time model for credit card fraud detection based on deep learning. In Proceedings of the 12th international conference on intelligent systems: theories and applications (pp. 1-7).
- Afriyie, J.K., Tawiah, K., Pels, W.A., Addai-Henne, S., Dwamena, H.A., Owiredo, E.O., Ayeh, S.A. and Eshun, J.**, 2023. A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions. *Decision Analytics Journal*, 6, p.100163.
- Aftabi, S.Z., Ahmadi, A. and Farzi, S.**, 2023. Fraud detection in financial statements using data mining and GAN models. *Expert Systems with Applications*, 227, p.120144.
- Alarfaj, F.K., Malik, I., Khan, H.U., Almusallam, N., Ramzan, M. and Ahmed, M.**, 2022. Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms. *IEEE Access*, 10, pp.39700-39715.
- Bagchi, D., Mukherjee, A. and Pal, S.**, 2021. A one step further approach to fraud detection. *Journal of Computer Science and Engineering (JCSE)*, 2(2), pp.112-119.
- Bagga, S., Goyal, A., Gupta, N. and Goyal, A.**, 2020. Credit card fraud detection using pipeling and ensemble learning. *Procedia Computer Science*, 173, pp.104-112.
- Carcillo, F., Le Borgne, Y.A., Caelen, O., Kessaci, Y., Oblé, F. and Bontempi, G.**, 2021. Combining unsupervised and supervised learning in credit card fraud detection. *Information sciences*, 557, pp.317-331.
- Carneiro, N., Figueira, G. and Costa, M.**, 2017. A data mining based system for credit-card fraud detection in e-tail. *Decision Support Systems*, 95, pp.91-101.
- Carta, S., Fenu, G., Recupero, D.R. and Saia, R.**, 2019. Fraud detection for E-commerce transactions by employing a prudential Multiple Consensus model. *Journal of Information Security and Applications*, 46, pp.13-22.
- Cherif, A., Badhib, A., Ammar, H., Alshehri, S., Kalkatawi, M. and Imine, A.**, 2022. Credit card fraud detection in the era of disruptive technologies: A systematic review. *Journal of King Saud University-Computer and Information Sciences*.

- de Sa, A.G., Pereira, A.C. and Pappa, G.L.,** 2018. A customized classification algorithm for credit card fraud detection. *Engineering Applications of Artificial Intelligence*, 72, pp.21-29.
- Dornadula, V.N. and Geetha, S.,** 2019. Credit card fraud detection using machine learning algorithms. *Procedia computer science*, 165, pp.631-641.
- Fanai, H. and Abbasimehr, H.,** 2023. A novel combined approach based on deep Autoencoder and deep classifiers for credit card fraud detection. *Expert Systems with Applications*, 217, p.119562.
- FBI,** “Federal Bureau of Investigation IC3 annual report”, <https://www.fbi.gov/contact-us/field-offices/springfield/news/internet-crime-complaint-center-releases-2022-statistics> (2022) (Erişim tarihi: 13.03.2023)
- Forough, J. and Momtazi, S.,** 2021. Ensemble of deep sequential models for credit card fraud detection. *Applied Soft Computing*, 99, p.106883.
- Geeks for geeks,** “Deep learning introduction to long short term memory”, <https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/> (2023) (Erişim tarihi: 14 Şubat 2023)
- Gupta, P., Varshney, A., Khan, M.R., Ahmed, R., Shuaib, M. and Alam, S.,** 2023. Unbalanced Credit Card Fraud Detection Data: A Machine Learning-Oriented Comparative Study of Balancing Techniques. *Procedia Computer Science*, 218, pp.2575-2584.
- Hilal, W., Gadsden, S.A. and Yawney, J.,** 2022. Financial fraud: a review of anomaly detection techniques and recent advances. *Expert systems With applications*, 193, p.116429.
- Islam, M.A., Uddin, M.A., Aryal, S. and Stea, G.,** 2023. An ensemble learning approach for anomaly detection in credit card data with imbalanced and overlapped classes. *Journal of Information Security and Applications*, 78, p.103618.
- JavaTPoint,** “Stacking in Machine Learning”, <https://www.javatpoint.com/stacking-in-machine-learning> (Erişim tarihi: 8 Haziran 2023).
- Karthikeyan, T., Govindarajan, M. and Vijayakumar, V.,** 2023. An effective fraud detection using competitive swarm optimization based deep neural network. *Measurement: Sensors*, 27, p.100793.

- Lei, Y.T., Ma, C.Q., Ren, Y.S., Chen, X.Q., Narayan, S. and Huynh, A.N.Q.,** 2023. A distributed deep neural network model for credit card fraud detection. *Finance Research Letters*, 58, p.104547.
- Machine learning mastery,** “A gentle introduction to Long Short-Term memory networks by the experts”, <https://machinelearningmastery.com/autoencoder-for-classification/> (2020) (Erişim tarihi: 7 şubat 2023)
- Machine learning mastery,** “Autoencoder feature extraction for classification”, <https://machinelearningmastery.com/autoencoder-for-classification/> (2020) (Erişim tarihi: 19 Kasım 2022)
- Makki, S.,** 2019. An efficient classification model for analyzing skewed data to detect frauds in the financial sector (Doctoral dissertation, Université de Lyon; Université Libanaise).
- Medium,** “Restricted Boltzmann machine (RBM) with practical implementation.”, <https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/> (2021) (Erişim tarihi: 28 Aralık 2022)
- Mehndiratta, S. and Gupta, K.,** 2019. Credit card fraud detection techniques: a review. *Int. J. Computer Sci. Mobile Computing*, 8(8), pp.43-49.
- Misra, S., Thakur, S., Ghosh, M. and Saha, S.K.,** 2020. An autoencoder based model for detecting fraudulent credit card transaction. *Procedia Computer Science*, 167, pp.254-262.
- Mittal, S. and Tyagi, S.,** 2019, January. Performance evaluation of machine learning algorithms for credit card fraud detection. In 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 320-324). IEEE.
- Nguyen, N., Duong, T., Chau, T., Nguyen, V.H., Trinh, T., Tran, D. and Ho, T.,** 2022. A proposed model for card fraud detection based on Catboost and deep neural network. *IEEE Access*, 10, pp.96852-96861.
- ProjectPro,** “A comprehensive guide to ensemble learning methods”, <https://www.projectpro.io/article/a-comprehensive-guide-to-ensemble-learning-methods/432> (2023) (Erişim tarihi: 19 Mart 2023)
- Pumsirirat, A. and Liu, Y.,** 2018. Credit card fraud detection using deep learning based on auto-encoder and restricted boltzmann machine. *International Journal of advanced computer science and applications*, 9(1).

- Quah, J.T. and Sriganesh, M.,** 2008. Real-time credit card fraud detection using computational intelligence. Expert systems with applications, 35(4), pp.1721-1732.
- Raghavan, P. and El Gayar, N.,** 2019, December. Fraud detection using machine learning and deep learning. In 2019 international conference on computational intelligence and knowledge economy (ICCIKE) (pp. 334-339). IEEE.
- Reshma, R.S.,** 2018. Deep learning enabled fraud detection in credit card transactions. Int J Res Sci Inn (IJRSI) V (VII).
- Roseline, J.F., Naidu, G.B.S.R., Pandi, V.S., alias Rajasree, S.A. and Mageswari, N.,** 2022. Autonomous credit card fraud detection using machine learning approach☆. Computers and Electrical Engineering, 102, p.108132.
- Rtayli, N. and Enneya, N.,** 2020a. Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization. Journal of Information Security and Applications, 55, p.102596.
- Rtayli, N. and Enneya, N.,** 2020b. Selection features and support vector machine for credit card risk identification. Procedia Manufacturing, 46, pp.941-948.
- Runchi, Z., Ligu, X. and Qin, W.,** 2023. An ensemble credit scoring model based on logistic regression with heterogeneous balancing and weighting effects. Expert Systems with Applications, 212, p.118732.
- Sahin, Y., Bulkan, S. and Duman, E.,** 2013. A cost-sensitive decision tree approach for fraud detection. Expert Systems with Applications, 40(15), pp.5916-5923.
- Shahana, T., Lavanya, V. and Bhat, A.R.,** 2023. State of the art in financial statement fraud detection: A systematic review. Technological Forecasting and Social Change, 192, p.122527.
- Somasundaram, A. and Reddy, S.,** 2019. Parallel and incremental credit card fraud detection model to handle concept drift and data imbalance. Neural Computing and Applications, 31(Suppl 1), pp.3-14.
- Van Belle, R., Baesens, B. and De Weerd, J.,** 2023. CATCHM: A novel network-based credit card fraud detection method using node representation learning. Decision Support Systems, 164, p.113866.

- Wu, B., Lv, X., Alghamdi, A., Abosag, H. and Alrizq, M., 2023.** Advancement of management information system for discovering fraud in master card based intelligent supervised machine learning and deep learning during SARS-CoV2. *Information Processing & Management*, 60(2), p.103231.
- Xu, B., Wang, Y., Liao, X. and Wang, K., 2023.** Efficient fraud detection using deep boosting decision trees. *Decision Support Systems*, p.114037.
- Zhang, X., Han, Y., Xu, W. and Wang, Q., 2021.** HOBA: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture. *Information Sciences*, 557, pp.302-316.



TEŞEKKÜR

Tez çalışmam boyunca değerli görüş ve önerileriyle doğru yönde ilerlememi sağlayan, her konuda yardım ve ilgisini esirgemeyen değerli hocam ve tez danışmanım Prof. Dr. Mustafa Serdar KORUKOĞLU hocama teşekkürlerimi sunarım. Tez izleme komitesi toplantılarındaki kıymetli görüş ve önerilerle çalışmama katkılar sağlayan Prof. Dr. Vecdi Aytaç ve Prof. Dr. Aytağ ONAN hocalarıma da ayrıca teşekkür ederim.

Evliliğimiz boyunca ve onun öncesinde, her konuda olduğu gibi akademik yaşamımda da bana sınırsız destek olan sevgili eşim Maryam AZARNIA'ya sonsuz teşekkür ederim.

Her zaman olduğu gibi tüm öğrenim hayatım boyunca beni destekleyen sevgili babam Abdolhalim BAGHDADI ve annem Tazehgol TOINOGLI'ya, teşekkürü bir borç bilirim. Ayrıca, annemden hiç farkı olmayan ve her zaman destekçi olup benim üvey annem Ejo ABDOLLAZADEH'ye de teşekkürlerimi sunarım.

Doktora sürecinde tanışmış olup kardeş gibi değerli fikirlerini ve yardımlarını esirgemeyen dostum Houtan VAFAEIFARD'a ayrıca teşekkürlerimi sunarım.

ÖZGEÇMİŞ

1. KİŞİSEL BİLGİLER

Ad Soyad:

Peyman BAGHDADI

2. EĞİTİM

Öğrenim Dönemi	Derece	Üniversite	Öğrenim Alanı
2019 - 2024	Doktora	Ege Üniversitesi	Bilgisayar Mühendisliği
2014 - 2016	Yüksek Lisans	Elmo Sanat Tehran Üniversitesi	Bilgisayar Mühendisliği
2007 - 2009	Lisans	Sepahan Yüksek Öğrenim Enstitüsü	Bilgisayar Mühendisliği
2005 - 2007	Ön Lisans	Elmo Sanat Arak Üniversitesi	Bilgisayar Yazılımı

3. İŞ DENEYİMİ

İş Dönemi	İş yeri	İş Ünvanı
2021 - ...	KentKart/Türkiye	Cihaz yazılım geliştirici uzmanı
2017 - 2018	Behpardakht (Mellat Bankası)/İran	terminal cihazı tasarım uzmanı
2016 - 2017	Ideanegar Energy Sabz/İran	gömülü sistem geliştiricisi
2015 - 2016	Ayandeh Banka/İran	ödeme sistemi ve POS cihazları geliştirici uzmanı
2012 - 2015	Kish enterprise/iran	ERP yazılım çözümünün geliştiricisi ve uzmanı
2011 - 2012	Tose-e danesh institute/iran	ICDL sertifika programının bilgisayar öğretmeni

4. YAYIN BİLGİLERİ

SCI veya SCI Expanded, SSCI, AHCI (WILEY) tarafından taranan dergilerde yayınlamak üzere submit edilen tam makale

1. Baghdadi, P., Korukoglu, S., Aytac, V., Onan, A., (2023) Implementation of a Decision Support System for Credit Card Fraud Detection Using Ensemble Deep Learning Models. *Cognitive Computation and Systems*.